NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMAL REPAIR AND REPLACEMENT POLICY FOR A SYSTEM WITH MULTIPLE COMPONENTS

by

Jan-Wilhelm Brendecke

June 2016

Thesis Co-Advisors:	David L. Alderson
	Kyle Y. Lin
Second Reader:	Michael P. Atkinson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE
06-17-2016

3. REPORT TYPE AND DATES COVERED
Master's Thesis 10-15-2015 to 06-16-2016

4. TITLE AND SUBTITLE
OPTIMAL REPAIR AND REPLACEMENT POLICY FOR A SYSTEM WITH MULTIPLE COMPONENTS

5. FUNDING NUMBERS

6. AUTHOR(S)
Jan-Wilhelm Brendecke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis formulates and solves a Markov decision problem to find the optimal repair and replacement policy for a system of multiple components whose failure rates are age-dependent. We assume that the failure rate for an old component is higher than for that of a new component. When a component fails, it can either be replaced, making it new, or repaired, making it functional but old. An old component can also be replaced proactively. We formulate the model for a single component as a linear program, and perform parametric analysis on the transition probabilities and system rewards to understand when different policies are optimal. We extend the model to include multiple, independent components, and apply the model to a notional infrastructure network whose performance depends on the state of its network links.

14. SUBJECT TERMS
Failure, Repair, Replace, Multiple Component Models, Maintenance Models, Markov Decision Process

15. NUMBER OF PAGES 83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

OPTIMAL REPAIR AND REPLACEMENT POLICY FOR A SYSTEM WITH MULTIPLE COMPONENTS

Jan-Wilhelm Brendecke
Major, German Army
Dipl.-Ing.(FH), University of Applied Sciences Koblenz, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2016

Approved by: David L. Alderson
Thesis Co-Advisor

Kyle Y. Lin
Thesis Co-Advisor

Michael P. Atkinson
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis formulates and solves a Markov decision problem to find the optimal repair and replacement policy for a system of multiple components whose failure rates are age-dependent. We assume that the failure rate for an old component is higher than for that of a new component. When a component fails, it can either be replaced, making it new, or repaired, making it functional but old. An old component can also be replaced proactively. We formulate the model for a single component as a linear program, and perform parametric analysis on the transition probabilities and system rewards to understand when different policies are optimal. We extend the model to include multiple, independent components, and apply the model to a notional infrastructure network whose performance depends on the state of its network links.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

NPS	Naval Postgraduate School
US	United States
MDP	Markov Decision Process
LP	Linear Program
IGFR	Increasing Generalized Failure Rate

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

This thesis aims to find an optimal repair and replacement policy for a system that consists of several components. At any time point, each component is either operational or non-operational. An operational component may fail and become nonoperational, and the failure rate increases as the component ages. The real-time performance of the system depends on the subset of components that are operational, and reaches its best when all components are operational. In order to maintain a high-level performance of the system in the long run, it is necessary to repair or replace non-operational components, or even to replace an old, but operational, component, before it fails. This thesis formulates and solves a mathematical model for this problem.

We begin by modeling the behavior of a single component, and categorize the realtime status of a component into three states: new, old, and failed. The component is operational in the first two states, and non-operational in the last state. As time goes by, a new component will either become old or fail at some point. The time in the new and old state are independent geometric random variables. An old component will eventually fail. If we choose to repair a failed component, then the component will become operational, but old. The remaining time to failure has a geometric distribution with same mean as before the failure. If we choose to replace a failed or an old component, then it will become new again. The state of a component is known at all times. We develop this model in the framework of a Markov decision process, and formulate a linear program to compute the optimal solution. The objective for the linear program is to minimize the long-run average cost of operating the component. This cost includes the operational costs, which depends on the state of the component and any cost for repairs and replacements. The optimal repair and replacement policy depends on several model parameters such as the component failure rates in different states, and the cost to repair or to replace a component.

We next extend the model to account for multiple components in the system. This can be done in a straightforward manner given our modeling framework, but the computational resources required to solve the problem grow very quickly. On a personal

computer, it takes several minutes to compute the optimal repair and replacement policy for a system with six components, but several hours to compute that for a system with seven components.

Our modeling framework makes it possible to consider several model variations, such as limiting the number of components that can be repaired or replaced at the same time, or studying the system performance if we cannot tell whether a component is new or old until it fails.

We demonstrate our model by applying it to a fuel network, which consists of nodes and links connecting them. Each node has either a supply or demand of fuel, and fuel is transported between nodes via the links. A link corresponds to a component in this system, and is subject to age-based failure. The system performs at its best when all links are operational, and the performance degrades with each link becoming nonoperational. We compute the optimal repair and replacement policy for a fuel network consisting of six links, and draw insight into the optimal repair and replacement policy via parametric analysis.

Our model is generic and applies to many real-world systems, such as fuel networks, transportation systems, and electricity grids. The model is flexible such that constraints can be added to the linear program to account for a variety of scenarios. The downside of our model is the computational burden required to solve it as the problem size increases. A future research direction is to develop efficient heuristic methods that can produce near-optimal policy with much less computational effort.

Acknowledgments

I would like to thank the German Army, since it was the Army's idea to send me here. I also would like to thank the Naval Postgraduate School for offering this great learning- and life-experience to international students.

A special thanks goes to my thesis advisors Professor David L. Alderson and Professor Kyle Y. Lin. I really enjoyed working with you and I'm very grateful for all your support during my thesis work.

And of course, I have to thank my parents. I would not be where I am today without you.

And last but not least, to all the great friends I made in my time here - thanks for all the shared memories and experiences. We had an awesome time here!

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Motivation

We are surrounded by various systems that influence our life in one way or another. Some of these systems make our lives more comfortable, while others are essential to preserve our current level of security or to maintain the daily operations of a military unit, a company, or even a country.

Regardless of the importance of a system, all systems are subject to failure. Most physical systems will degenerate over time, and some may fail due to a shock like an accident or an attack. Typically, a system becomes more likely to fail as it ages. One way to mitigate the age-based risk of failure is to replace a system proactively, when the risk of age-based failure becomes too high. Although there are various models in the literature that address the issue of aged-based failure, most of these models focus on a single component or on a system as a single entity.

In modern days, most complicated systems consist of more than just one component. Each of these individual components may fail independently. A system's performance may depend not only on the number of operational and non-operational components, but also on the configuration of which components are operational and which are not. Motivated by this observation, this thesis aims to find an optimal repair and replacement policy for a system that consists of several components, where each component may fail independently and the overall system performance depends on the operating states of its components.

This simple characterization describes many systems. One such example is a network, which moves commodity—such as fuel, water, or electricity—between nodes via links. The links correspond to the components in our model. Each link can either be operational or not. The performance of such a system depends on the set of links that are operational. Alderson et al. (2015) introduce a mathematical formulation for such a network, where they study how to optimize system performance when a subset of the links become nonoperational. This thesis complements that work by allowing the links to degrade and fail, and studies when to repair and replace these links to improve system resilience.

Government officials need to reinforce the relevance of critical infrastructures—such as a fuel network—and the importance of the resilience of such infrastructures (see The White House 2015). Resilience describes, on the one hand, the critical infrastructure's ability to withstand damages, and, on the other hand, its ability to remain operational in the event of a failure. This stresses the importance of an optimal repair and replacement policy for such systems.

1.2 Overview

This thesis studies when to repair or replace individual components in a complicated system in order to improve system resilience. We aim for a model that describes a generic system of multiple independent components which are prone to age-based failure. Each of these components has individual parameters to describe their behavior. The system reward depends on the overall state of all components. The component parameters and the reward structure can be modified to capture the behavior of various systems.

Chapter 2 reviews the related literature on models and methods for optimal repair and replacement policies. Chapter 3 introduces a model for a single component that is subject to aging and failure. We develop a model in the framework of a Markov decision process, and formulate a linear program to compute the optimal repair and replacement policy. We apply this model to a canonical example of single-component, and we conduct parametric analysis to gain insight to the behavior of this model. In Chapter 4 we extend the single-component model to a model with multiple components. We then introduce and examine possible modifications and extensions to model more complex scenarios of a multiple-component model. In Chapter 5 we apply the model to a notional fuel network as formulated by Alderson et al. (2015). We then examine the optimal repair and replacement policy, conduct parametric analyses and present modifications to the model to capture different scenarios. Finally, Chapter 6 concludes our works and offers a few recommendations.

CHAPTER 2:
Literature Review

There is a rich literature on systems with components that are subject to failure. For example, Barlow and Hunter (1961) consider reliability of a single component system.

Derman (1963b) developed a model for optimal replacement policies if the change of state is Markovian. Derman (1963a) discusses mathematical optimization techniques for replacement policies.

McCall (1965), Pierskalla and Voelker (1976), and Sherif and Smith (1981) provide surveys of maintenance models for systems with components that fail stochastically. Agrawal and Barlow (1984) provide a survey of network reliability models.

This chapter reviews selected works relevant to the type of system under study. We begin with a review of key modeling features, and then consider previous work using Markov decision processes to study optimal policies of repair and replacement of aging components.

2.1 Dimensions of the Problem

There are a variety of repair and replacement models that have been studied in the literature. Each of these models considers different features and exposes different tensions and tradeoffs in model behavior. As a background to our model, we examine some of the dimensions of these problems here.

Discrete vs. Continuous Time. For all models that incorporate time, there is always the decision between discrete time and continuous time. Discrete time models use fixed time steps or time periods to measure time. Multiple events can occur during one time step, but change in the system only occurs at the end of that time step. The resolution depends on the size of the time-step. In general, a model has more detail if it uses smaller time steps. However, the size of the time steps should be meaningful in context of the total modeled time span. Love et al. (2000) and von Noortwijk and Frangopol (2014) give examples of discrete-time maintenance models.

Continuous-time models do not divide the time in fixed blocks. An event can happen at any given time point and the system can change its status at any time point. Dogramaci and Fraiman (2004) give an example of a continuous-time maintenance model.

Age-Dependent Failure Rates. One key assumption for all maintenance-related models is that the failure depends on the age of the system. The implementation of this behavior depends on if the age of the system is modeled in discrete time or continuous time. For continuous-time maintenance models, the time to the next failure is a random value that depends on the age of the system and decreases with age. The probability of failure increases with age. Therefore, continuous time models need an associated lifetime distribution that captures that behavior. The technical term for this type of behavior is *Probability Distributions with Increasing Generalized Failure Rates* (IGFR). The Gamma and Weibull distributions are examples for IGFR. Lariviere (2006) explains methods to show that a distribution is in fact IGFR. Discrete-time models specify the probability that a system fails in the current time step. In general, the probability of failure increases with each consecutive time step. Pierskalla and Voelker (1976) and Ross (2014) show various models that use age-depended failure rates for discrete- and continuous-time models.

Shock-Based Failures. Unlike age-dependent failure rates, some discrete- and continuous-time models use shock-based failures. These models assume that some external events or shocks influence the durability of the component. For example, a shock could be an attack, an accident, or just the regular use of that component. In models of shock-based failure, the probability of failure increases with the number of shocks endured. The geometric distribution is widely used to determine the number of shocks a system can sustain before failure. The Poisson distribution is often used to describe the number of shocks that happen over a certain time. A shock-based model is introduced by Zhang (2002).

Perfect vs. Imperfect Repair. All maintenance-related models in the literature have some mechanism to address a failed system. Some consider a perfect or good-as-new repair, while others consider an imperfect repair. A *perfect repair* will reset a failed component to be a brand new component. An *imperfect repair* will ensure that the system is operational again, but the state of the system is not that of new one. The probability of failures is reset to that of a component of a certain age or certain number of shocks. This offset can be fixed, or it can be a random value. Models use imperfect repair to include the notion that a system can not be repaired indefinitely. See Zhang (2002) for an example of perfect repair and Love et al. (2000) for an example of imperfect repair.

Repair vs. Replace. A different notion of prefect and imperfect repair in the same system is to model repair and replacement separately. The replacement of a failed system introduces a brand new component to the system. Therefore, this new component has a lower failure rate than the original component. The repair of a failed component does not introduce a new component. The state of the component changes from failed to an operational state. The age or number of shocks respectively remain unchanged. Therefore, the failure rate of that component is the same as before. The replacement is more costly than the repair of a failed component. Repair and replace models address the tradeoff between higher cost and lower failure rates. See Pierskalla and Voelker (1976) for examples of such models.

Inventory Constraints. Another dimension of maintenance models is the question of how to address the supply of spare parts and that of new components. Most models in the literature consider an unlimited supply of these parts (e.g. Pierskalla and Voelker 1976). Other models require a certain availability of some sort of resources and parts to conduct any maintenance action. The cost of buying and storing resources are also included in these models. Different maintenance actions, such as repair and replace, often require different amounts or types of resources; therefore the cost of these actions differ. Rajagopalan (1998) models a system with such constraints.

Maintenance, Minor and Major Repair. Sim and Endrenyi (1993) differentiate between maintenance, minor repair and major repair. In this context, maintenance is conducted on the operational system. Minor and major repairs are conducted if the system has failed. Maintenance slightly decreases the probability of failure of the operational component. Without maintenance, the wear and tear leads to a failure. The wear and tear is modeled by continuously increasing the probability of failure over time. If the system fails, one can decide between minor and major repair. Both types of repair will ensure that the system is operational again. A minor repair uses the least amount of resources, e.g. parts, time, money, possible to achieve operational readiness. A major repair resets the system to the best state possible. Minor and major repairs are similar to repair and replace in that aspect. McCall (1965) provides additional examples of such models.

Proactive Replacement. In contrast to regular replacement, proactive replacement is conducted while the system is still operational. Sherif and Smith (1981) examine models which use proactive replacement. The probability of failure increases with the age of the system. To decrease the probability of failure, an operational component is replaced with a new component if it reaches a certain age. The tradeoff is that a proactive replacement is more costly than doing nothing but less costly than a repair or replacement after a failure.

Complete vs. Incomplete Information. A model that supports state-based decision making requires that the underlying state is known. It is possible to replace an old component only if the age of that component is actually known. In the literature, we find models with two different levels of information. One level of information is to know if a system is operational or has failed. This level of information is considered the base level, since it is easy to distinguish between operational and non-operational. In contrast, a maintenance action on an operational system requires information about that precise state of the system. The age or number of shocks of the system must be known to decide if a replacement is necessary. The former cases are known as cases of *incomplete information*, while the latter cases are known as cases of *complete information*. Pierskalla and Voelker (1976) give examples for models with complete and incomplete information In Thomas et al. (1987) the state of the system is only known when it is inspected.

2.2 Markov Decision Processes

A Markov decision process (MDP) is a mathematical framework used to address decision problems where system behavior is partly random and partly the result of actions by a decision maker. A key assumption is that of the Markov property, namely that the future state of the system depends on the current state only and not on past states. The reward of a system depends on the time spent in each state, and the core problem of an MDP is to find a policy for the decision maker that maximizes total reward.

Ross (2014) provides an introduction to MDPs. A more detailed explanation can be found in Puterman (2005). They show that to define a Markov decision process, we need to define its state space, action space, the transition probability associated with each state-action pair, and the cost associated with each state-action pair.

To minimize the long-run average cost per time period, we can formulate a linear program. The decision variables represent the long-run fraction of time for each state-action pair. The objective function is the linear combination of long-run proportion of time for the state-action pairs and its corresponding reward. The constraints are flow-balance constraints, where the left-hand side is the long-run fraction of transitions leaving a state, and the right-hand side is the long-run fraction of transitions entering a state. See chapter 9.3 of Hillier and Lieberman (2010) for an example formulation of such a linear program.

The theory of a MDP states that, for our type of model, there exists an optimal policy. To find this policy we solve the linear program. The solution shows that for each state only one state-action pair, represented by a decision variable, will be non zero. This implies the optimal action for that state. By examining all non zero decision variables we find the optimal action for each state and therefore the optimal policy for the system

2.3 Our Problem in Context

We aim to find the optimal repair and replacement policy for a system of multiple components. In the context of the aforementioned problem dimensions, we restrict attention to the following.

1. We model age-based failure rates instead of shock-based failure rates.
2. We differentiate between repair and replacement. In this context, we understand repair as an imperfect repair. A repair action will change a component from non-operational to operational, but the probability of failure will be higher

compared to that of a new component. A replacement action will change a component from non-operational to operational, and the component will have the same probability of failure as a new component.

3. We allow proactive replacement. A proactive replacement is the replacement of a component that is operational but not new.
4. We assume complete information. The exact state of all components is known at any time. The model parameters are also known.
5. We use discrete time instead of continuous time.
6. We do not incorporate inventory constraints.

We use a Markov decision process to find the optimal repair and replacement policy for a single component system, and then for a system of multiple components. The complexity of the model depends on two factors: the number of components and the number of valid state-action pairs in the Markov decision process. Although we do not consider explicit inventory constraints, we do consider cases where the number of available workers limits the repairs or replacements that can be performed in any time period. Since our model is time-discrete, we assume that a repair and replacement both require one time-step each. A time-continuous model would allow for more flexibility and could capture various repair and replacement times. We favor the less flexible discrete-time model, because it allows the use of a very small state-space. A small state-space is necessary to extend the single component model to a multiple component model.

CHAPTER 3:
A Single Component Model

This chapter introduces a mathematical model to study the optimal maintenance policy of a system whose components are subject to age-based failure rates. We begin with a single-component model in Section 3.1, and extend it to a system of multiple components in Chapter 4.

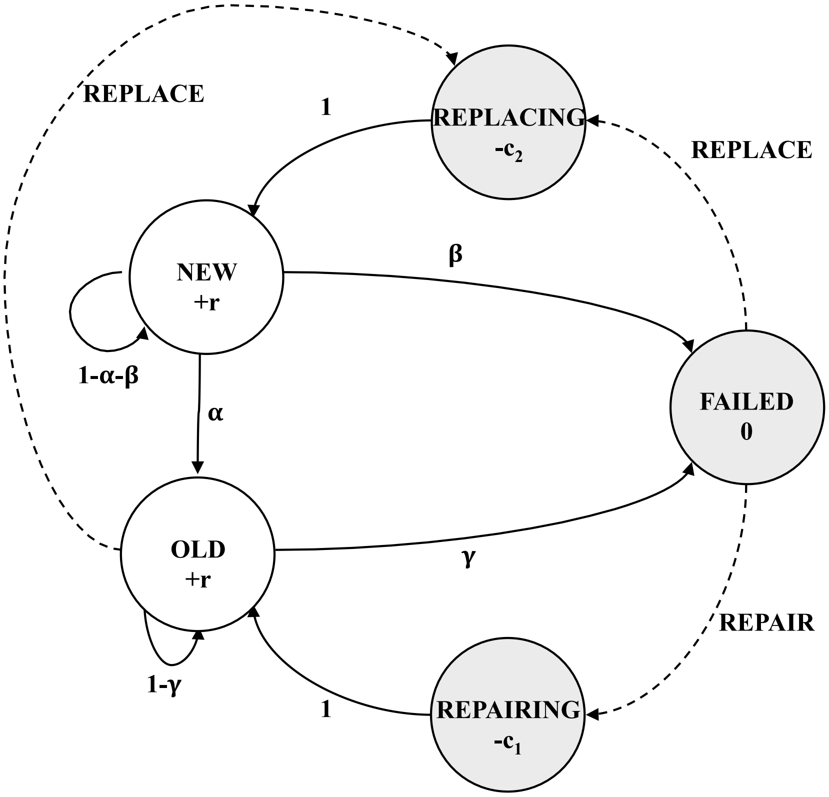
3.1 Model Description

Consider a system that consists of a single component. Time is discrete. In each time period, the component is in one of the following five *states*:

1. NEW: The component is as good as new, with a relatively low failure probability.
2. OLD: The component is still *operational*, but has a higher failure probability.
3. FAILED: The component has failed and is *non-operational*.
4. REPAIRING: The component is under repair, and is therefore non-operational. Its state will become OLD after the completion of the repair.
5. REPLACING: The component is being replaced and is therefore non-operational. Its state will become NEW after the completion of the replacement.

Figure 3.1 shows the possible transitions between these states. Whenever in state NEW, in the next time period, the component may becomes OLD with probability α, or become FAILED with probability β, or remain NEW with probability $1-\alpha-\beta$. Whenever in state OLD, in the next time period, the component may become FAILED with probability γ, or remain OLD with probability $1-\gamma$. Whether in states NEW or OLD, the component is operational.

When the component is FAILED, an *action* needs to be taken to change its state; otherwise it will remain FAILED in the next time period. There are two such actions. If we choose to *repair* a FAILED component, then the state becomes REPAIRING in the next time period, and then OLD in the following time period. In other words, it takes one time period to repair a component, and a repaired component is operational



A component is in one of five possible states. It can transition to another state according to a transition probability (solid lines) or due to an action taken by the operator (broken lines). The non shaded states are considered operational states with a positive reward r. The shaded states are considered non-operational. Replacing a component incurs cost c_1 while repairing incurs cost c_2

Figure 3.1: The Single Component Model

but not as good as new. If we choose to *replace* a FAILED component, then the state becomes REPLACING in the next time period, and then NEW in the following time period. In other words, it also takes one time period to replace a component. The component is nonoperational in states FAILED, REPAIRING, or REPLACING.

As seen in Figure 3.1, when the component is in state OLD, one can choose to proactively replace the component before it fails. Whether it is wise to do so, however, depends on the cost parameters, which will be discussed in the next section.

3.2 Linear Program Formulation

This model can be formulated as a Markov Decision Process. To do so, we specify the set of possible actions that apply to each state, and then a reward structure associated with each state-action pair. We then formulate a linear program to compute the optimal policy that maximizes the long-run average reward per time period.

In each state, there are up to three possible *actions*:

1. NONE: This action applies to all five states.
2. REPAIR: This action repairs a FAILED component.
3. REPLACE: This action replaces a FAILED component or an OLD component.

Altogether, there are eight meaningful state-action pairs, which are summarized and enumerated in Table 3.1. Please note that the action REPAIR is applicable in state FAILED, but not applicable in state REPAIRING. If the action REPAIR is applied to state FAILED, then the component will transition to state REPAIRING in the next time period, and the state will become OLD in the following time period. For the same reason, the action REPLACE is applicable in states OLD or FAILED, but not applicable in state REPLACING. It is valid to apply action NONE to state FAILED. In a single component system this is only possible if the cost for repair and replacement are extremely high compared to the reward.

Table 3.1: Eight Meaningful State-Action Pairs

	NONE	REPAIR	REPLACE
NEW	1	—	—
OLD	2	—	6
FAILED	3	7	8
REPAIRING	4	—	—
REPLACING	5	—	—

When the component is operational (states NEW or OLD), a reward r is received for each time period, whereas no such reward is received when the component is nonoperational (states FAILED, REPAIRING, or REPLACING). There is a cost c_1 to repair a component, and a cost c_2 to replace a component. Note that in the case of proactive replacement (i.e., REPLACE when the component is in state OLD, the component will only be non-operational for one time step. If the REPLACE action happens while the component is in state FAILED, the component is non-operational for two time-steps. In the first time-step it transitions to the state FAILED and can only move to the state REPLACING in the next time step.

We next formulate a linear program to maximize the long-run average profit (reward less cost) per time period. This is a standard procedure to solve a Markov decision process. See Hillier and Lieberman (2010) Chapter 19.1 for a detailed explanation. Let x_j denote the long-run fraction of time for state-action pair j, $j = 1, \dots, 8$. These x_j are the decision variables in the linear program, whose values will imply a policy. Formally, we solve the following linear program.

$$\max_{x_1,\dots,x_8} \quad r(x_1 + x_2 + x_6) - c_1 x_4 - c_2 x_5 \tag{3.1}$$

$$x_1 = x_5 + (1 - \alpha - \beta)\, x_1 \tag{3.2}$$

$$x_2 + x_6 = \alpha\, x_1 + x_4 + (1 - \gamma) x_2 \tag{3.3}$$

$$x_3 + x_7 + x_8 = \beta\, x_1 + \gamma\, x_2 + x_3 \tag{3.4}$$

$$x_4 = x_7 \tag{3.5}$$

$$x_5 = x_6 + x_8 \tag{3.6}$$

$$x_j \geq 0 \qquad j = 1, \dots, 8 \tag{3.7}$$

$$\sum_{j=1}^{8} x_j = 1 \tag{3.8}$$

The objective function in Equation (3.1) is the long-run average reward less cost per time period. Constraints (3.2) to (3.6) are flow balance constraints, one for each state. The left-hand side is the flow out of the state, while the right-hand side is the flow into the state. Constraint (3.7) requires all eight variables to be nonnegative, since they represent long-run fraction of time. Finally, constraint (3.8) ensures that in each time period, the process corresponds to one and only one state-action pair.

3.3 Numerical Demonstration

To implement the linear program, we use the Python Programming Language (PSF 2016) with the Pyomo optimization modeling language (Hart et al. 2012, 2011), and CPLEX optimizer (IBM 2016) to solve it. We present a numerical example. Consider the following transition probabilities

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4,$$

and the following reward and cost parameters

$$r = 5, \qquad c_1 = 1, \qquad c_2 = 2.$$

For these parameters, the solver returns the optimal value 2.913, which is the maximized long-run average profit (reward less cost) per time period. Below is the optimal solution (rounded to three digits), with the corresponding state-action pair noted in the parentheses:

$$\begin{aligned}
x_1 &= 0.290 \qquad (\text{NEW}, \text{NONE})\\
x_2 &= 0.362 \qquad (\text{OLD}, \text{NONE})\\
x_3 &= 0.000 \qquad (\text{FAILED}, \text{NONE})\\
x_4 &= 0.000 \qquad (\text{REPAIRING}, \text{NONE})\\
x_5 &= 0.174 \qquad (\text{REPLACING}, \text{NONE})\\
x_6 &= 0.000 \qquad (\text{OLD}, \text{REPLACE})\\
x_7 &= 0.000 \qquad (\text{FAILED}, \text{REPAIR})\\
x_8 &= 0.174 \qquad (\text{FAILED}, \text{REPLACE})
\end{aligned}$$

We can derive the optimal decisions from these optimal values. There are two decisions to be made, namely what to do in state FAILED and what to do in state OLD. In state FAILED, we need to decide among three actions NONE, REPAIR, or REPLACE. Because

$$x_3 = x_4 = 0, \qquad x_5 = 0.174,$$

it follows that the optimal decision is REPLACE in state FAILED. In state OLD, we need to decide between two actions NONE or REPLACE. Because

$$x_2 = 0.362, \qquad x_6 = 0,$$

it follows that the optimal decision is NONE in state OLD (i.e to not REPLACE an OLD component before it fails).

To summarize, we operate a component when it is either NEW or OLD to earn reward. When the component fails, we replace it with a new one.

3.4 A Closed Form Solution

For a single component model, it is possible to enumerate all feasible policies and compute the long-run average reward for each of them. The optimal policy is the feasible policy that yields the highest long-run reward rate. Recall that when the component is in state OLD, there are two possible actions: NONE or REPLACE. When the component is in state FAILED, there are three possible actions: NONE or REPAIR or REPLACE. Therefore, the total number of feasible policies is $2 \times 3 = 6$. Below we evaluate these 6 feasible policies using a renewal reward process, so that the optimal solution can be expressed in a closed form.

Policy 1: OLD – NONE, FAILED – REPAIR.
With this policy, the state will cycle through OLD, FAILED, REPAIRING. Call it a renewal whenever the process enters state REPAIRING. The number of time periods in state OLD in a cycle follows a geometric distribution with parameter γ, so its expected value is $1/\gamma$. Each cycle also consists of 1 time period in state FAILED and 1 time period in state REPAIRING, so the expected cycle time is $1/\gamma + 2$. Since the reward is r for each time period in state OLD, and the repair cost c_1 is incurred once in each cycle, the long-run average reward is

$$\frac{\frac{1}{\gamma}r - c_1}{\frac{1}{\gamma} + 2}.$$

Policy 2: OLD – NONE, FAILED – REPLACE.

Call it a renewal whenever the process enters state REPLACING. In each cycle, the number of time periods in state NEW follows a geometric distribution with parameter $\alpha + \beta$. When the process leaves state NEW, it will enter state OLD with probability $\alpha/(\alpha + \beta)$, in which case the component will be in state OLD for a random number of time periods following a geometric distribution with parameter γ. Each cycle also consists of exactly 1 time period in state FAILED and 1 time period in state REPLACING. The long-run average reward is therefore

$$\frac{\frac{1}{\alpha+\beta}r + \frac{\alpha}{\alpha+\beta}\frac{1}{\gamma}r - c_2}{\frac{1}{\alpha+\beta} + \frac{\alpha}{\alpha+\beta}\frac{1}{\gamma} + 2}.$$

Policy 3: OLD – REPLACE, FAILED – REPLACE;
Call it a renewal whenever the process enters state REPLACING. In each cycle, the number of time periods in state NEW follows a geometric distribution with parameter $\alpha + \beta$. When the process leaves state NEW, whether to states OLD or FAILED, the action REPLACE will be taken in the following time period, so the expected cycle time is $1/(\alpha + \beta) + 1 + 1$. In addition, with probability $\alpha/(\alpha + \beta)$ a cycle will include 1 time period in state OLD, earning a reward r, so the long-run average reward is

$$\frac{\frac{1}{\alpha+\beta}r + \frac{\alpha}{\alpha+\beta}r - c_2}{\frac{1}{\alpha+\beta} + 2}.$$

Policy 4: OLD – REPLACE, FAILED – REPAIR.
Call it a renewal whenever the process enters state REPLACING. The number of time periods in state NEW follows a geometric distribution with parameter $\alpha + \beta$, so its expected value is $1/(\alpha + \beta)$. When the process leaves state NEW, it will enter state OLD with probability $\alpha/(\alpha + \beta)$, or enter state FAILED with probability $\beta/(\alpha + \beta)$. Since the policy is to REPAIR in state FAILED, each cycle consists of 1 time period each in states FAILED and REPAIRING with probability $\beta/(\alpha + \beta)$. Each cycle also includes exactly 1 time period in states OLD and REPLACING, respectively. Therefore, the long-run average reward is

$$\frac{\frac{1}{\alpha+\beta}r - \frac{\beta}{\alpha+\beta}c_1 + r - c_2}{\frac{1}{\alpha+\beta} + \frac{\beta}{\alpha+\beta} + \frac{\beta}{\alpha+\beta} + 2}.$$

Table 3.2: Feasible Policies

Policy	OLD	FAILED
1	NONE	REPAIR
2	NONE	REPLACE
3	REPLACE	REPLACE
4	REPLACE	REPAIR
5	NONE	NONE
6	REPLACE	NONE

Policy 5: OLD – NONE, FAILED – NONE.

Once the component reaches state FAILED, it will stay in state FAILED thereafter, so the long-run average reward is 0.

Policy 6: OLD – REPLACE, FAILED – NONE.

Once the component reaches state FAILED, it will stay in state FAILED thereafter, so the long-run average reward is 0.

Table 3.2 summarizes these six feasible policies. The optimal policy is the one that produces the highest long-run average reward.

Polices 1, 2 and 3 are the most intuitive policies, with Policy 1 being the most conservative and Policy 3 the most aggressive. Proactive replacement of an OLD component in Policy 3 has the advantage of keeping the component operational for an additional time step (i.e., only a single period of "downtime", vice two periods of "downtime" during replacement of a FAILED component). Two parameters drive the decision between active and proactive replacement, specifically the failure probability in state NEW and the failure probability in state OLD. If failure of an old component is very unlikely compared to that of a new component, the benefit from proactive replacement will be small. If the failure of an old component is much more likely than that of a new component, then the proactive replacement will be beneficial. Policy 4 is rather unintuitive, since when the component is in state FAILED, we will REPAIR the component to have it in state OLD in the next time period (earning r in that time period), and then immediately REPLACE it in the following time period to have a NEW component. Policy 5 and 6 are the trivial policies, with a long-run average reward of 0. These two policies correspond to the scenario, where the repair and replacements are very high relative to the reward, to the point that it is better

Table 3.3: Policy Rewards for several Transition Probabilities

Probabilities			Rewards			
α	β	γ	Policy 1	Policy 2	Policy 3	Policy 4
0.40	0.10	0.15	**3.7308**	3.7143	3.0000	2.9091
0.65	0.10	0.45	2.3947	**2.7183**	2.7000	2.6481
0.40	0.45	0.50	2.2500	2.6000	**2.6364**	2.5385
0.30	0.50	0.65	1.8643	1.8846	1.8913	**1.9167**
0.30	0.45	0.60	**2.0000**	**2.0000**	**2.0000**	**2.0000**

For all cases $r = 5, c_1 = 1, c_2 = 2$. The optimal policy is emphasized.

to shut down the component altogether. It is also possible to have ties among these policies. Table 3.3 shows examples of transition probabilities for which one of the non trivial polices is optimal and one example where all polices result in the same reward.

For a single component model, both this method and the computational method in Chapter 3.2 produce the same optimal solutions. However, for a system with multiple components, the number of feasible policies grows quickly, so the computation method in Chapter 3.2 becomes the only viable approach.

3.5 Parametric Analysis of the Single Component Model

The single-component model has six parameters: the three reward parameters (r, c_1, and c_2) and the three transition probabilities (α, β, and γ). This section presents parametric analysis on how these parameters affect the optimal policy. We consider only the nontrivial policies and exclude policies 5 and 6 from the analyses.

3.5.1 Parametric Analysis on Transition Probabilities

We perform parametric analysis on the three transition probabilities α, β, and γ. To do so, we pick the following rewards parameters

$$r = 5, \qquad c_1 = 1, \qquad c_2 = 2.$$

Although the only formal constraint is that α, γ, β each reside in the closed interval $[0, 1]$, we focus on the case $\beta < \gamma$, since in reality an OLD component is more likely to fail in the next time period than does a NEW component. In addition, we require $\alpha + \beta \leq 1$, since $1 - \alpha - \beta$ represents the probability that a NEW component will remain in state NEW for another time period, which must be nonnegative.

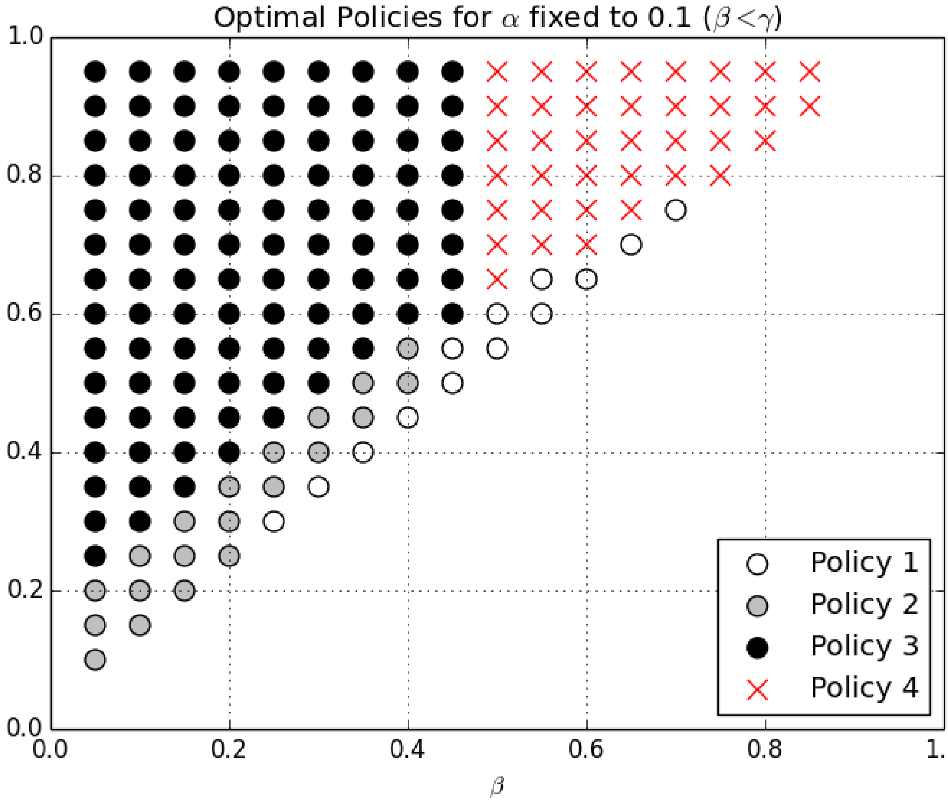


Figure 3.2: Optimal Policy based on β, γ for a fixed $\alpha = 0.1$, when $r = 5$, $c_1 = 1$ and $c_2 = 2$

Figure 3.2 depicts the optimal policy, when we set $\alpha = 1$ and vary β and γ. There are several interesting observations. First, Policy 1 (repair a FAILED component so the component is never NEW) is optimal only when $\beta \approx \gamma$ in certain areas. Since the quality of states NEW and OLD are comparable, when a component fails, it is better to repair it for a smaller cost c_1 rather than to replace it for a larger cost c_2. Second, for small β, as γ increases, the quality of an OLD component decreases, so the optimal policy becomes gradually more aggressive, from Policy 2 (replace in state FAILED) to Policy 3 (replace in states FAILED and OLD). Note that we fixed $\alpha = 0.1$, so the component is very likely to stay in state NEW. The failure probability for a component in state NEW is smaller compared to that of state OLD. Therefore it is beneficial to remain in state NEW as long as possible (i.e., take action REPLACE very often to remain in state NEW). This leads to a dominance of Policy 3. Third, for large β, as γ increases, the unintuitive Policy 4 (repair a FAILED component, use OLD component for one time period, and then immediately replace the OLD component with a NEW one) becomes optimal.

Since Policy 4 is rather unintuitive and unlikely to be optimal in a realistic scenario, we examine one of the scenarios from this analysis closer. We pick following parameters:

$$\alpha = 0.1, \quad \beta = 0.6, \quad \gamma = 0.8, \quad r = 5, \quad c_1 = 1, \quad c_2 = 2.$$

From the closed form solutions we know, that the long-run average reward rates for the 6 policies are

$$1.6154, \quad 1.6733, \quad 1.7083, \quad 1.8056, \quad 0, \quad 0,$$

respectively. Policy 4 has the highest average reward per time step.

In this example, since $\alpha + \beta = 0.7$, the NEW component most likely will become either OLD or FAILED in the next time period. Furthermore, an OLD component has a very high probability $\gamma = 0.8$ to become FAILED in the next time period. It turns out that in state FAILED, it is optimal to REPAIR it since $c_1 = 1$ is relatively cheap to ensure the component will be functional (state OLD) in the next time period. In addition, as soon as we use the OLD component for one time period, it is optimal to immediately replace it with a NEW component, since without doing so the OLD component is likely to fail (probability $\gamma = 0.65$) anyway, which will cause one extra time period of down time.

Figure 3.3 depicts the optimal policy, when we set $\beta = 0.1$ and vary α and γ. When γ is small and α is large, a component in state OLD tends to last for a while before becoming FAILED, and a component in state NEW tends to become OLD very quickly, so Policy 1 is optimal (repair in state FAILED). If we fix α, then as γ

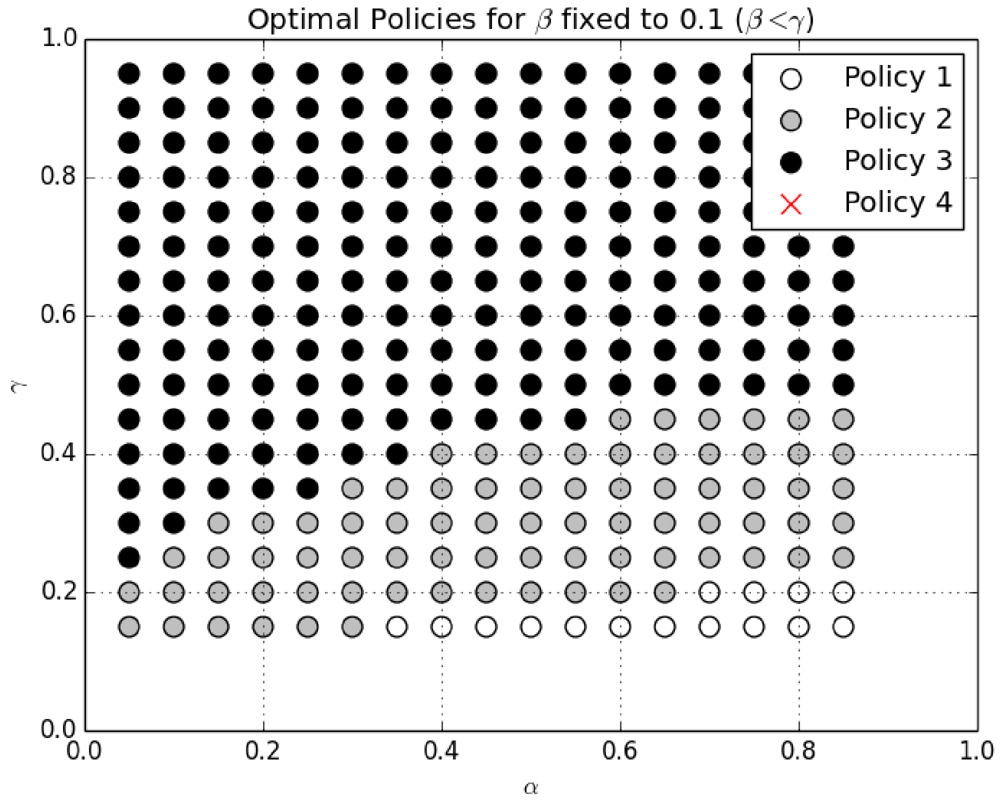


Figure 3.3: Optimal Policy based on α, γ for a fixed $\beta = 0.1$, with $r = 5$, $c_1 = 1$ and $c_2 = 2$

increases, the quality of state OLD decreases, so the optimal policy becomes more aggressive (from Policy 1 to Policy 2 to Policy 3).

Figure 3.4 depicts the optimal policy, when we set $\gamma = 0.5$ and vary α and β. When α or β is small, state NEW has great quality, so an aggressive policy to replace in either states OLD or FAILED is optimal (Policy 3). As either α increases or β increases, the quality of state NEW becomes closer to that of state OLD, so a less aggressive policy becomes optimal, namely Policy 2 and then Policy 1.

We also tested several different sets of reward values r, c_1, and c_2. Although the optimal policy depends on specific model parameters, the structure of the optimal policy observed in Figures 3.2 to 3.4 remains the same.

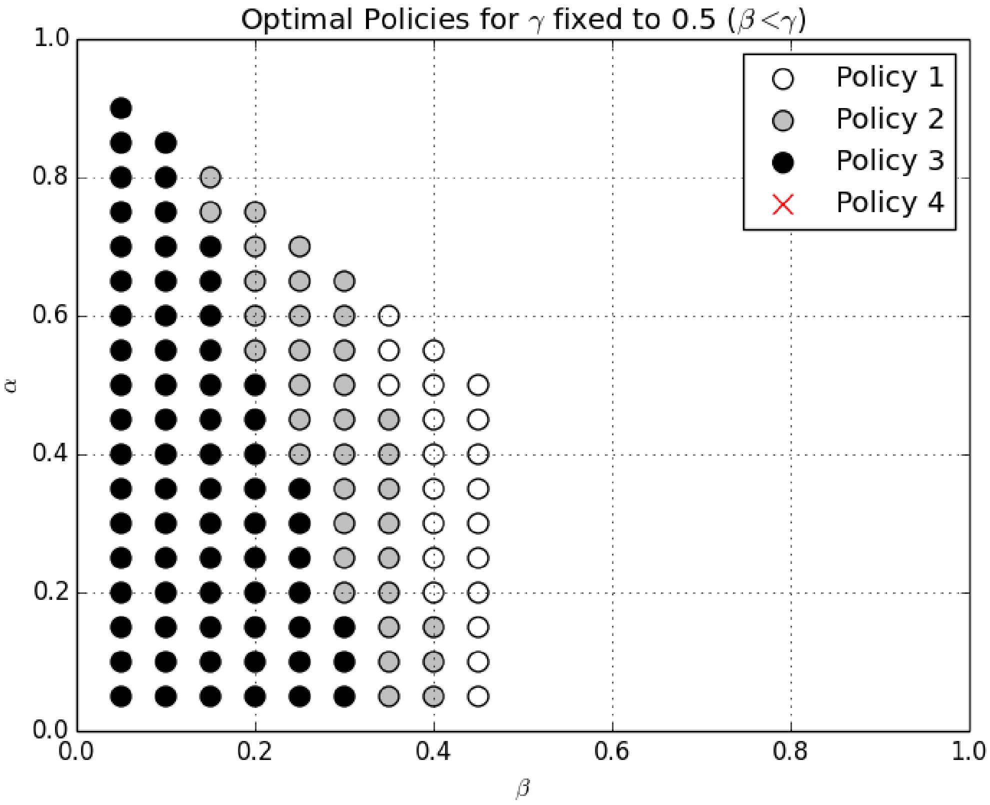


Figure 3.4: Optimal Policy based on α, β for a fixed $\gamma = 0.5$, with $r = 5$, $c_1 = 1$ and $c_2 = 2$

3.5.2 Parametric Analysis on Reward Structure

Next we perform parametric analysis on the three reward parameters r, c_1 and c_2. To do so we pick the following transition probabilities:

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4.$$

While we vary r, c_1, c_2 between 1 and 10, we focus on the case $c_1 < c_2$, since we assume that REPAIRING a component is less expensive than REPLACING it.

Figure 3.5 depicts the optimal policy, when we set $r = 6$, and vary c_1 and c_2. In almost all cases, the optimal policy is either Policy 1 or Policy 2. In other words, the optimal action is NONE in state OLD. The only case where it is optimal to replace in state OLD is $(c_1, c_2) = (1, 1.5)$—very small replacement cost. In addition, whether

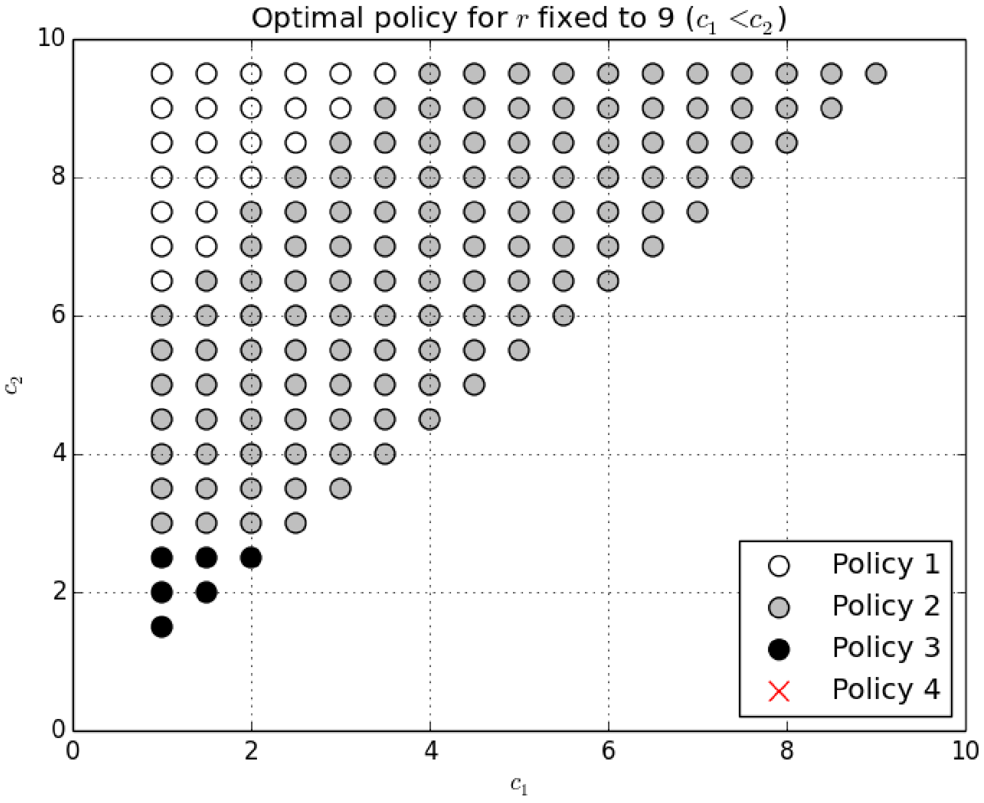


Figure 3.5: Optimal policy based on c_1, c_2 for fixed $r = 9$, with $\alpha = 0.5$, $\beta = 0.1$ and $\gamma = 0.4$

Policy 1 or Policy 2 is optimal largely depends on the ratio c_1/c_2. Policy 2 becomes a more attractive policy (replace in state OLD), either when c_1 (repair cost) increases, or when c_2 (replacement cost) decreases.

Figure 3.6 depicts the optimal policy when we set $c_1 = 1$, while varying r and c_2. In all cases, the optimal policy is either Policy 1 or Policy 2; in other words, the optimal action in state OLD is NONE. For a given value of r, as c_2 increases, the replacement cost becomes more expensive, so it becomes more attractive to simply repair in state FAILED, therefore Policy 1. For a given value of c_2, as r increases, it becomes more important to keep the component operational as much as possible, so it becomes more beneficial to replace in state FAILED, therefore Policy 2. These two structural properties can be clearly seen in Figure 3.6.

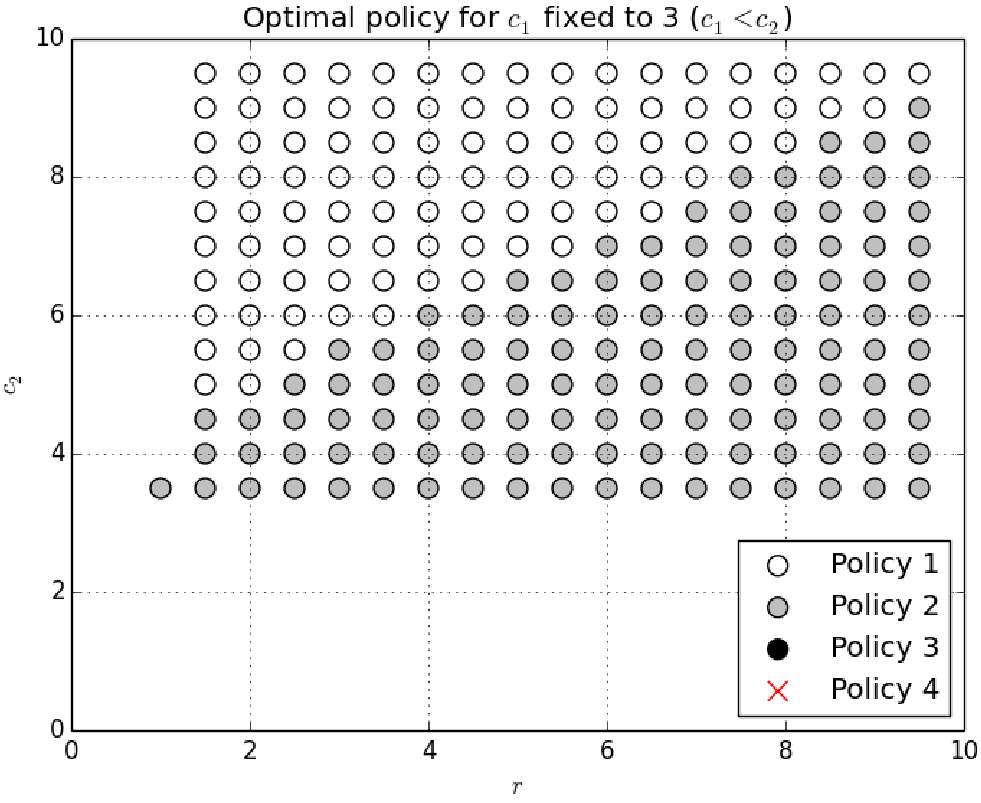


Figure 3.6: Optimal Policy based on r, c_2 for fixed $c_1 = 3$, with $\alpha = 0.5$, $\beta = 0.1$ and $\gamma = 0.4$

Figure 3.7 depicts the optimal policy when we set $c_2 = 7$, while varying c_1 and r. First, when r is very small, it becomes possible that the optimal solution is to shut down the component altogether, if c_1 is sufficiently large, leaving an optimal long-run average reward of 0. In Figure 3.7, a few trivial cases of this nature show up for $r = 1$ or $r = 1.5$, where an optimal policy is not shown. Second, in all cases shown, the optimal policy is either Policy 1 or Policy 2. In other words, the optimal action in state OLD is NONE, and the decision is whether to repair (Policy 1) or replace (Policy 2) in state FAILED. For a given value of c_1, as r increases, it becomes more important to keep the component operational as much as possible, so a more aggressive maintenance policy, namely Policy 2, becomes optimal. For a given value of r, as c_1 decreases, the repair cost drops, so it becomes more attractive to repair the component in state FAILED, therefore Policy 1. These two structural properties

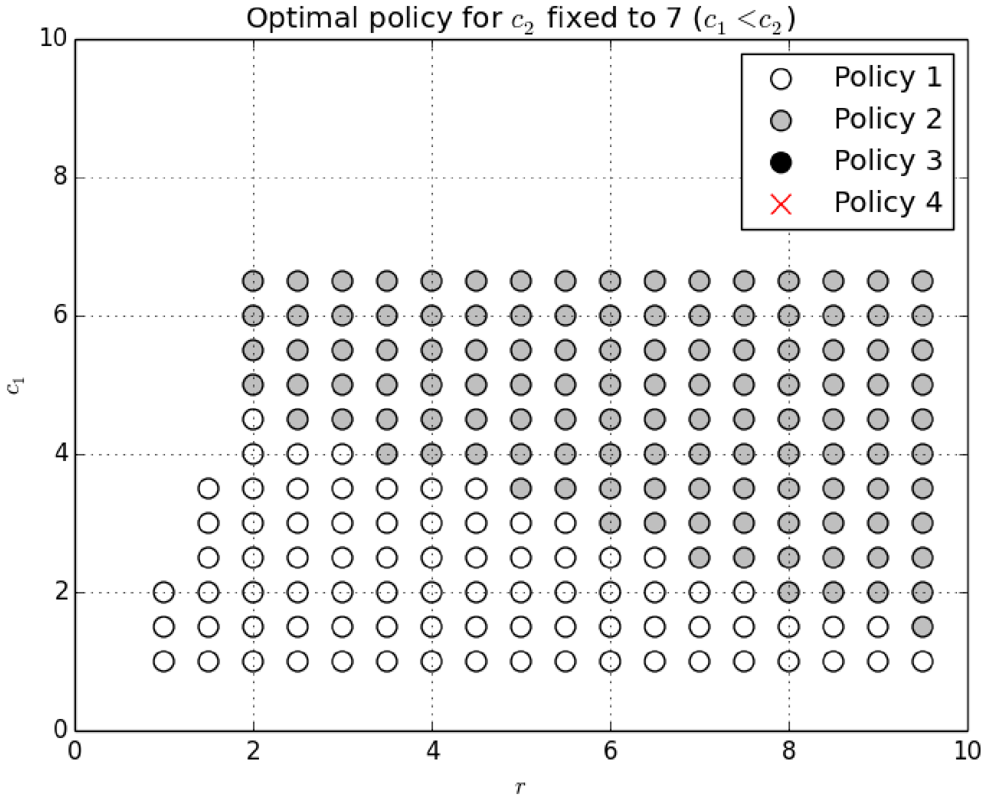


Figure 3.7: Optimal Policy based on r, c_1 for fixed $c_2 = 7$, with $\alpha = 0.5$, $\beta = 0.1$ and $\gamma = 0.4$

can be clearly seen in Figure 3.7.

In all scenarios we observe some intuitive structural properties. Different regions are occupied by distinct optimal policies, which implies that the optimal policy is of a threshold type, based on the values of the model's six parameters: $\alpha, \beta, \gamma, r, c_1, c_2$.

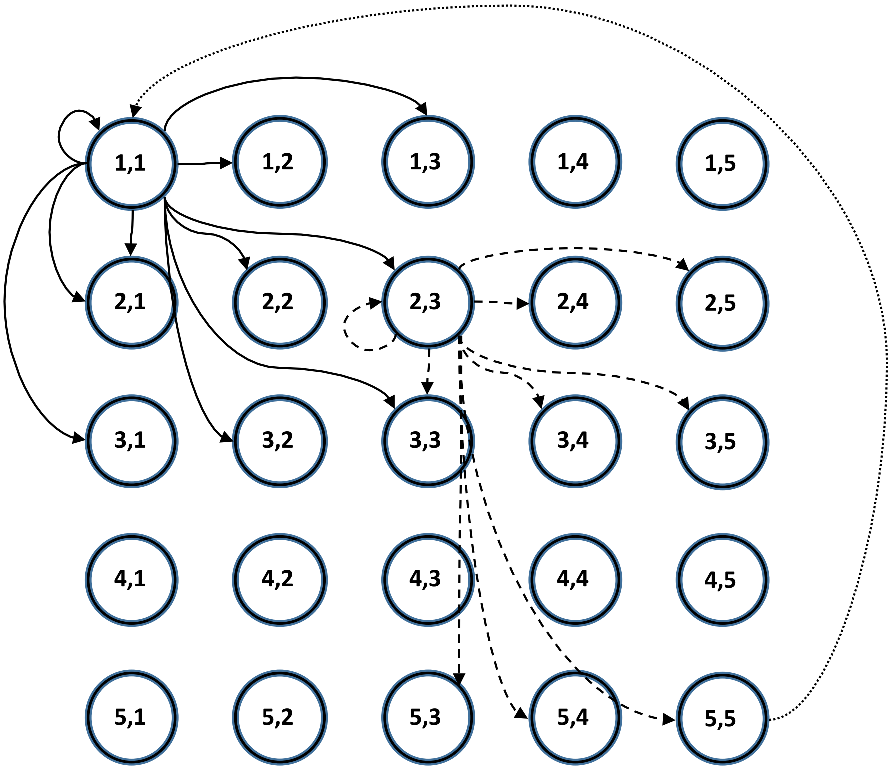
CHAPTER 4:
A Multiple Component System

This chapter introduces a model for a system with $n \geq 1$ components, where each component behaves the same way as described in Chapter 3. Each component deteriorates (from state NEW to state OLD) and fails (from state NEW to state FAILED or from state OLD to state FAILED) independently. We allow heterogeneous components, and write α_i, β_i and γ_i for the transition probabilities for component i, $i = 1, \ldots, n$.

4.1 Linear Program Formulation

A system with multiple components can be modeled by a Markov decision process, similar to the approach for a single-component system described in Section 3.2. We write the state of the system as $(s_1, s_2, \ldots, s_n)$, where s_i is the state of component i, for $i = 1, \ldots, n$. Since each component belongs to one of the five possible states shown in Table 3.1, there are a total of 5^n states. For example, a 2-component system is in state $(3, 1)$, if component 1 is in state FAILED and component 2 is in state NEW. There are 25 feasible states for such a system.

The optimal policy for a system with multiple components can be solved by a linear program, similar to the linear program formulated for the single-component model. The decision variables are the valid state-action pairs. Recall from Table 3.1 that each component has 8 valid state-action pairs, so the total number of decision variables is 8^n. Write $x_{j_1,j_2,\cdots,j_n}$ for the long-run fraction of time for the state-action pair $(j_1, j_2, \ldots, j_n)$, where $j_i \in \{1, 2, \ldots, 8\}$, for $i = 1, \ldots, n$. The state-action pairs for each component are as in Table 3.1 For example, in a system with $n = 2$ components, the decision variable $x_{2,7}$ stands for the long-run fraction of time that component 1 is in state OLD and the action is NONE, and component 2 is in state FAILED and the action is REPAIR. This model has no limitations on how many actions can be taken in each time step Figure 4.1 depicts the state space and selected transitions for a system with $n = 2$ components.



In a Two-Component Model the system state are the combination of the two component states. The Figure shows exemplary transitions for the states (1,1) in solid lines, (2,3) in dashed lines and (5,5) in doted lines.

Figure 4.1: State-Space of a Two-Component Model

The system performance depends directly on the state of the system. Suppose that the reward is $R(j_1 \dots j_n)$ for state-action pair $(j_1 \dots j_n)$. The objective function is to maximize the long-run reward per time period. The linear program is therefore

$$\max \sum_{(j_1 \cdots j_n)} R(j_1, j_2, \dots, j_n) \times x_{j_1, j_2, \dots, j_n}, \tag{4.1}$$

$$\text{s.t. flow balance constraints} \tag{4.2}$$

$$\sum x_{j_1, \dots, j_n} = 1, \tag{4.3}$$

$$x_{j_1, \dots, j_n} \geq 0, \qquad \text{for all } (j_1, \dots, j_n) \tag{4.4}$$

Since there are 5^n states, there are 5^n flow-balance constraints in Equation (4.2), one for each state. For example, in a system with $n = 2$ components, there are $5^2 = 25$

flow-balance constraints. The flow-balance constraint for state $(1,1)$ is

$$x_{1,1} = (1-\alpha_1-\gamma_1)(1-\alpha_2-\gamma_2)x_{1,1} + (1-\alpha_1-\gamma_1)x_{1,5} + (1-\alpha_2-\gamma_2)x_{5,1} + x_{8,8}, \tag{4.5}$$

where the left-hand side is the flow leaving state $(1,1)$ and the right-hand side is the flow entering state $(1,1)$. If both components are in state NEW, then there is a probability of $(1-\alpha_1-\gamma_1)(1-\alpha_2-\gamma_2)$ that both of them will be NEW again in the next time period. If component 1 is in state NEW and component 2 is in state REPLACING, then there is a probability $(1-\alpha_1-\gamma_1)$ that both components will be NEW in the next time period, since component 2 will be NEW with probability 1. If both components are in state REPLACING, then with probability 1 both components will be in state NEW in the next time period. The flow-balance constraints for the other states can be formulated analogously.

Below we present an example with $n = 2$ components. Suppose that each component has transition probabilities

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4.$$

In addition, the action REPAIR costs $c_1 = 1$ and the action REPLACE costs $c_2 = 2$. The reward function is

$$R(j_1, j_2) = 5 \times \text{the number of operational components.}$$

In other words, each operational component (states NEW or OLD) generates a reward 5 per time period.

For these parameters, the linear program returns the optimal value 5.826, which is the maximized long-run average profit (reward less cost) per time period. Below is the optimal solution, with the corresponding state-action pairs noted in the parentheses. We show only the nonzero decision variables, rounded to three digits. A complete printout of the optimal solutions can be found in Appendix A.

$x_{1,1} = 0.084$	(NEW, NONE), (NEW, NONE)
$x_{1,2} = 0.105$	(NEW, NONE), (OLD, NONE)
$x_{1,5} = 0.050$	(NEW, NONE), (REPLACING, NONE)
$x_{1,8} = 0.050$	(NEW, NONE), (FAILED, REPAIR)
$x_{2,1} = 0.174$	(OLD, NONE), (NEW, NONE)
$x_{2,2} = 0.131$	(OLD, NONE), (OLD, NONE)
$x_{2,5} = 0.063$	(OLD, NONE), (REPLACING, NONE)
$x_{2,8} = 0.063$	(OLD, NONE), (FAILED, REPAIR)
$x_{5,1} = 0.050$	(REPLACING, NONE), (NEW, NONE)
$x_{5,2} = 0.063$	(REPLACING, NONE), (OLD, NONE)
$x_{5,5} = 0.030$	(REPLACING, NONE), (REPLACING, NONE)
$x_{5,8} = 0.030$	(REPLACING, NONE), (FAILED, REPLACE)
$x_{8,1} = 0.050$	(FAILED, REPLACE), (NEW, NONE)
$x_{8,2} = 0.063$	(FAILED, REPLACE), (OLD, NONE)
$x_{8,5} = 0.050$	(FAILED, REPLACE), (REPLACING, NONE)
$x_{8,8} = 0.050$	(FAILED, REPLACE), (FAILED, REPLACE)

For each component, the only nonzero state-action pairs are (NEW, NONE), (OLD, NONE), (FAILED, REPLACE), and (REPLACING, NONE). Thus, for each component, the optimal solution is to leave it alone until it fails, then replace it.

When comparing these results with those from the single-component model in Section 3.3, we observe that each individual component of the two-component model preserves the same optimal action as the component in the single-component model. Additionally, the optimal solution of each decision variable in the two-component model matches the product of the corresponding optimal solutions in the single-component model. These results can also be understood intuitively, since both com-

Table 4.1: Parameters for Different Multiple-Component Models

# of Components	# of Decision Variables	# of Constraints	Runtime (sec)
1	8	5	0.023
2	64	25	0.030
3	512	125	0.084
4	4096	625	0.720
5	32768	3125	9.439
6	262144	15625	310.274
7	2097152	78125	26214.107

The runtimes have been measured on a personal Computer with a 2.5 GHz Intel Core i7 Processor and 16 GB 1600 MHz DDR3 RAM.

ponents are identical and operate independently. This extension to a two-component model is useful conceptually to validate our logic and to automate the formulation of the linear program. However, our interest is in the formulation and solution of a more general system with n components.

Our linear program formulation works for a system with n components, for arbitrary value of n. However, the number of variables and the number of constraints both grow rapidly as n grows. For a system with n components, there are 5^n states and 8^n state-action-pairs, so the linear program has 8^n decision variables and 5^n flow-balance constraints. Table 4.1 shows how the numbers of decision variables and flow balance constraints grow as n increases. The times to run the linear program on a personal computer are shown in the last column. We consider a seven-component model as the largest system that can be solved on a personal computer in a reasonable time.

4.2 Additional Constraints: Limited Workers

Suppose that the action REPAIR requires 1 worker, while the action REPLACE requires 2 workers. Moreover, assume there are a limited number of available workers. In this case, only some of the system states are feasible.

Below we demonstrate the idea by a system consisting of $n = 2$ identical components. Each of these components has the following parameters:

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4, \qquad r = 5, \qquad c_1 = 1 \qquad c_2 = 2.$$

In addition, suppose that we have a total of 2 workers available in each time period. In other words, it is infeasible to repair one component and replace the other (which requires 3 workers), or to replace both components (which requires 4 workers), in the same time period. As seen in Table 4.2, actions $(4, 5)$, $(5, 4)$, and $(5, 5)$ are infeasible, so we can set

$$x_{4,5} = x_{5,4} = x_{5,5} = 0$$

in the linear program to enforce this requirement. A complete printout of these results is in appendix A.

Table 4.2: Required Workers per Action

Action	# of Workers Required
(1,4) (2,4) (3,4) (6,4) (7,4) (8,4)	1
(1,5) (2,5) (3,5) (6,5) (7,5) (8,5) (4,4)	2
(4,5) (5,4)	3
(5,5)	4

By setting the number of available workers to different values, we can compare how the system performance changes, as shown in Figure 4.2. Since in our 2-component model, the maximal number of workers needed is $2 \times 2 = 4$, we are interested in comparing the system performance for the number of workers ranging from 1 to 4. With 0 worker, both components will be in state FAILED permanently, so the long-run reward rate is 0. With more than 4 workers, the long-run reward rate is the same as the case with 4 workers.

As seen in Figure 4.2, the long-run reward rate increases as the number of workers increases. This result is intuitive, since with more workers, more actions become feasible. With only one worker, the only feasible action is to repair one failed component. If both components are in state FAILED, the system needs an additional time period to repair the second component. It is also not possible to replace a failed component with just one worker.

In addition, Figure 4.2 shows that the marginal improvement of the long-run reward rate decreases as the number of workers increases. This result also makes intuitive sense, since most often the optimal solution requires 1 or 2 workers, and rarely does it require 3 or 4 workers. In particular, the only situation where 4 workers may be

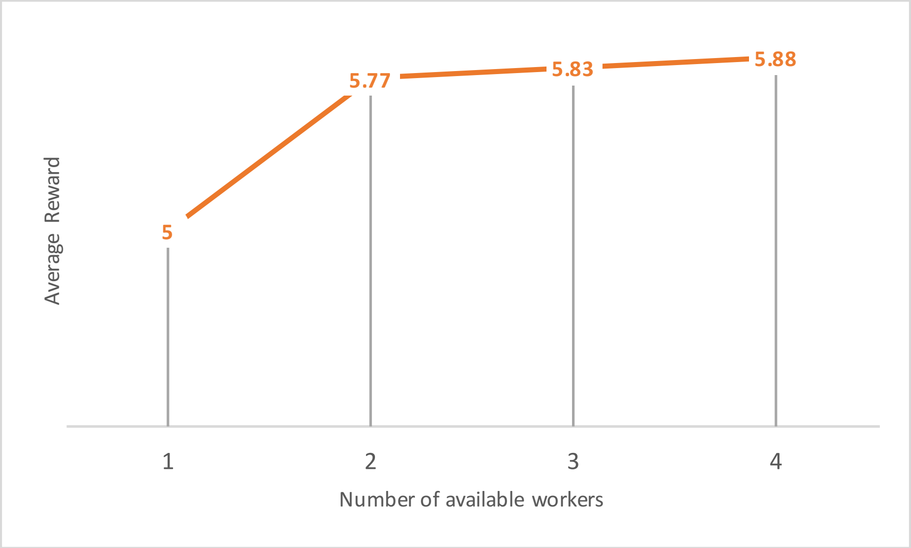


Figure 4.2: Reward for Different Number of Workers

needed is when both components are in state FAILED, which occurs very infrequently. Hence, the improving going from 3 workers to 4 workers is very little.

4.3 Additional Variations

The linear program in Section 4.1 can be modified to account for a variety of real-world scenarios. Here are a few examples:

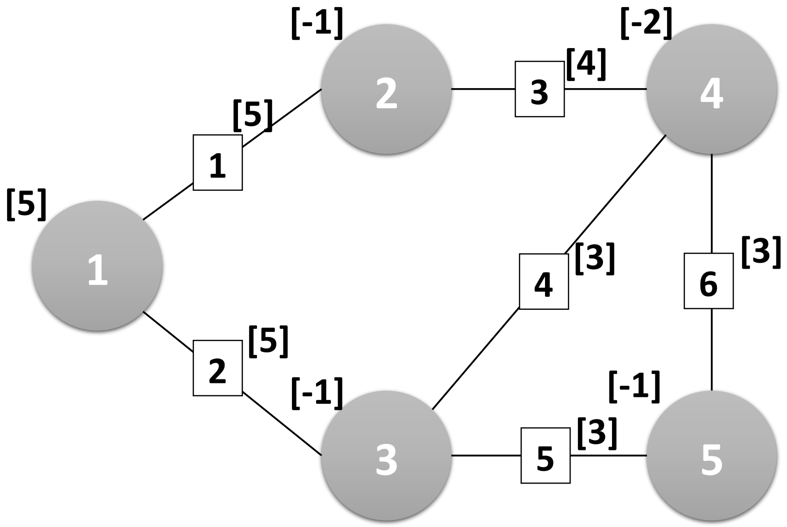
1. If the action REPLACE requires a certain machine, and we only have y such machines available, then we can set all decision variables that replace more than y components in the same time period equal to 0.
2. If we cannot tell whether the state of a component is states NEW or OLD when it is operational, and consequently do not want to allow the action REPLACE in state OLD, then we can set all corresponding decision variables to 0.
3. If the cost for multiple REPLACE actions in the same time step is cheaper per component, then for a single component in one time step (e.g., discount, fixed cost), then we can modify the reward structure accordingly.

The value of each feature depends on the application of interest. The next chapter applies the multiple-component model to a notional infrastructure system of practical interest.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5: Application to a Fuel System

In this chapter, we apply the multiple component model to the study of a notional fuel infrastructure system, as defined by Alderson et al. (2015). Figure 5.1 shows a example of such a fuel network with five nodes and six links. Each node has either a supply or demand of fuel, represented by positive or negative value respectively. Fuel is transported between nodes via the links, each of which has a maximum capacity and cost per unit of fuel that is transported via this edge.



The nodes (circles) are numbered from 1 to 5. The number in brackets next to each node is the supply at that node (a negative number indicates a demand). The links are numbered from 1 to 6. The number next to these numbers is capacity of that link. The cost to transport one unit over one edge is 1. The per-unit penalty for failing to deliver fuel to a demand location is 10.

Figure 5.1: A Six-Component Fuel System

We adopt the minimum-cost flow network formulation defined in Alderson et al. (2015). This model assumes that the system operator faces two types of costs: flow delivery costs (for the movement of fuel from supplies to demands) and penalty costs (for each unit of unsatisfied demand). The overall objective of the operator is to minimize this aggregate cost under each scenario.

We now apply the multiple component model to find the optimal repair and replacement policy for this notional fuel system. We assume that only the links are subject to failure. Because the fuel system has six edges, we have a six-component model.

5.1 Multiple-Component Model Setup

Our model requires the following input data:

- reward structure of the system,
- repair and replacement costs for each component, and
- transition probabilities for each component.

The starting point for our analysis is a complete reward structure for the network under each in each possible combination of component (link) states. From the perspective of fuel delivery, a link that is NEW or OLD is simply "operational" while a link that is FAILED, REPAIRING, or REPLACING is "non-operational." We use the model from Alderson et al. (2015) to compute the operating cost for the system. For any combination of operational and non-operational links, the model finds the least costly solution, including both delivery costs and non-delivery penalties. Table 5.1 lists the system performance associated with each of the $2^6 = 64$ link combinations.

On top of the raw system performance, we additionally add the repair and replacement cost to each decision variables containing the action REPAIR or the action REPLACE. We must complete the procedure for all $8^6 = 262,144$ decision variables. Figure 5.2 shows an example of this process. The resulting reward structure is a list with 262,144 entries. Due to the large number of entries we refrain from showing the list here.

We pick following parameters for the components or edges respectively and assume that the edges are identical in respect to these parameters,

$$\alpha = 0.2, \quad \beta = 0.1, \quad \gamma = 0.3, \quad c_1 = 1, \quad c_2 = 2.$$

Further, we introduce the number of workers required to complete one REPAIR action or REPLACE action. We denote these as w_1 and w_2 respectively. For the base

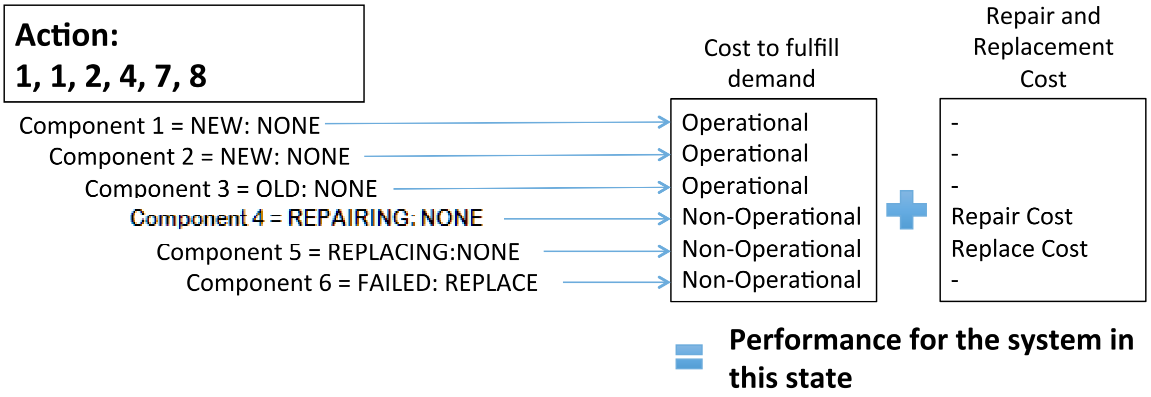
Table 5.1: Performance of a Fuel System under each possible Link Configuration.

Link i state						System	Link i state						System
1	2	3	4	5	6	Performance	1	2	3	4	5	6	Performance
0	0	0	0	0	0	8	1	0	0	0	0	0	10
0	0	0	0	0	1	8	1	0	0	0	0	1	18
0	0	0	0	1	0	9	1	0	0	0	1	0	18
0	0	0	0	1	1	16	1	0	0	0	1	1	18
0	0	0	1	0	0	8	1	0	0	1	0	0	19
0	0	0	1	0	1	8	1	0	0	1	0	1	33
0	0	0	1	1	0	9	1	0	0	1	1	0	41
0	0	0	1	1	1	16	1	0	0	1	1	1	41
0	0	1	0	0	0	8	1	0	1	0	0	0	17
0	0	1	0	0	1	8	1	0	1	0	0	1	17
0	0	1	0	1	0	9	1	0	1	0	1	0	18
0	0	1	0	1	1	16	1	0	1	0	1	1	25
0	0	1	1	0	0	10	1	0	1	1	0	0	19
0	0	1	1	0	1	24	1	0	1	1	0	1	33
0	0	1	1	1	0	32	1	0	1	1	1	0	41
0	0	1	1	1	1	32	1	0	1	1	1	1	41
0	1	0	0	0	0	11	1	1	0	0	0	0	50
0	1	0	0	0	1	12	1	1	0	0	0	1	50
0	1	0	0	1	0	11	1	1	0	0	1	0	50
0	1	0	0	1	1	18	1	1	0	0	1	1	50
0	1	0	1	0	0	12	1	1	0	1	0	0	50
0	1	0	1	0	1	25	1	1	0	1	0	1	50
0	1	0	1	1	0	18	1	1	0	1	1	0	50
0	1	0	1	1	1	25	1	1	0	1	1	1	50
0	1	1	0	0	0	41	1	1	1	0	0	0	50
0	1	1	0	0	1	41	1	1	1	0	0	1	50
0	1	1	0	1	0	41	1	1	1	0	1	0	50
0	1	1	0	1	1	41	1	1	1	0	1	1	50
0	1	1	1	0	0	41	1	1	1	1	0	0	50
0	1	1	1	0	1	41	1	1	1	1	0	1	50
0	1	1	1	1	0	41	0	1	1	1	1	0	50
0	1	1	1	1	1	41	1	1	1	1	1	1	50

For each link i, 0 indicates "operational" and 1 indicates "non-operational." System performance is computed according to the minimum-cost flow model defined in Alderson et al. (2015).

scenario, we do not put a limitation on the number of available workers. We pick the following values:

$$w_1 = 1, \quad w_2 = 2.$$



The action "1, 1, 2, 4, 7, 8" translates to component 1 is in state NEW, action is NONE; component 2 is in state NEW, action is NONE, component 3 is in state OLD, action is NONE, component 4 is in state REPAIRING, action is NONE, component 5 is in state REPLACING, action is NONE, component 6 is in state FAILED, action is REPLACE. Therefore components 1, 2 and 3 are operational. components 4, 5 and 6 are non-operational. The performance of the fuel system in this state is 16 (see Table 5.1). Since components 4 and 5 are being repaired or replaced we have to add the cost c_1 and c_2 , respectively. The final performance of the system in this state is $16 + 1 + 2 = 19$

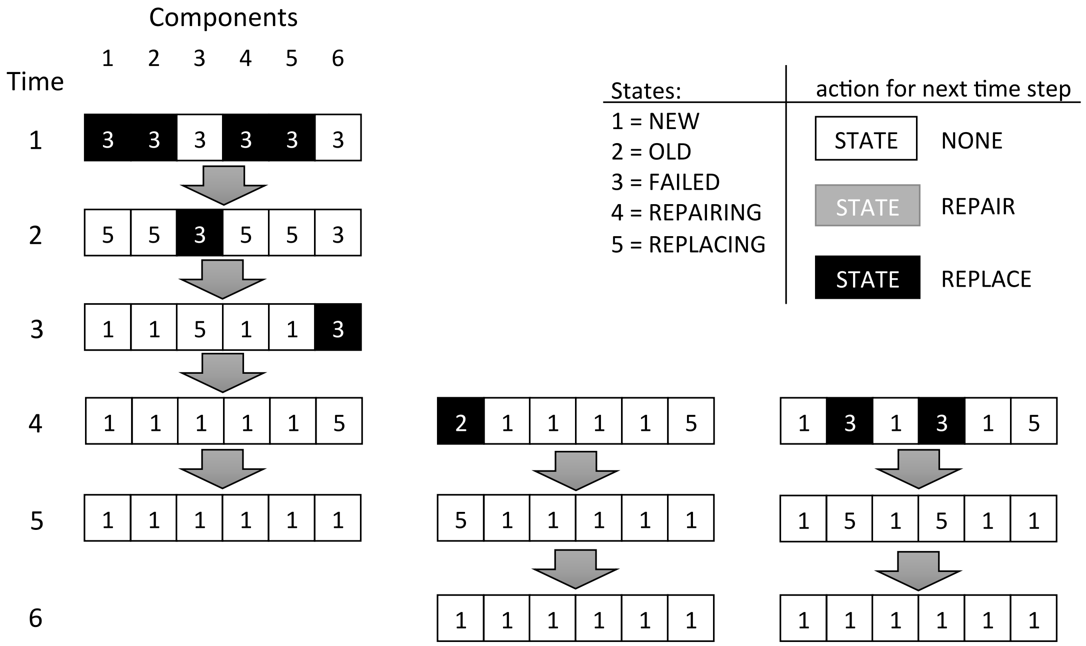
Figure 5.2: The Reward Mapping Process for a Fuel System

5.2 Optimal Repair and Replacement Policy

With all of the necessary inputs, we can solve the linear program defined in Section 4.1. In the case of the fuel system we want to minimize the long-run average cost of moving fuel from the supply nodes to the demand nodes. Therefore our objective function (5.1) is the sum over the long-run average proportion of time the system is one of these states multiplied with the cost associated for that state.

$$\min \sum_{(j_1 \cdots j_{262144})} R(j_1, j_2, \ldots, j_{262144}) \times x_{j_1, j_2, \ldots, j_{262144}} \tag{5.1}$$

Recall that the non-zero decision variables imply the optimal repair and replacement policy for this fuel system. For every possible combination of the components states an optimal action exists. We examine the optimal policy for two illustrative cases.



Optimal repair and replacement actions, along with resulting states, when all six components are initially in state FAILED. Left column: the optimal replacement schedule in the absence of any other events. Center column: if component 1 transitions to state OLD, it is optimal to proactively replace it. Right column: if components 2 and 4 transition to state FAILED, then these need to be replaced.

Figure 5.3: Optimal Policy if all Components are FAILED

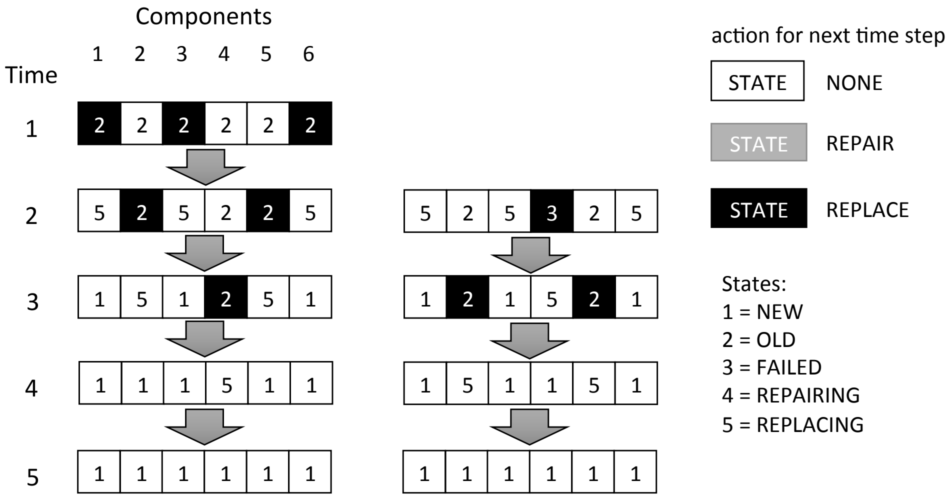
Case 1: all components are FAILED

Consider the case where initially all six components are in state FAILED. Figure 5.3 shows the optimal policy and consecutive states starting from this initial state. In the first time step the optimal action is to replace components 1, 2, 4 and 5. Since there is no limitation on the number on actions per time step, it seems intuitive to replace all components immediately. However, the structure of this network (see Figure 5.1) is that we can satisfy all demands if just these four components are operational. Moreover, because a component can only transition once per time step, a component that just transitioned from state REPLACING to state NEW will remain in state NEW for at least one time step. By spreading the REPLACE action over multiple time steps the system reduces the probability of multiple transitions from state NEW to state OLD or state FAILED in one time step. Following this policy, all components are operational in time step 5.

The decision to repair or replace a component does not preclude transitions by other components. Specifically, other components can transition to state OLD or state FAILED, creating multiple branches in the sequence of states that result from any particular policy. Figure 5.3 depicts two other possible branches at time step 4. In the center column, the component 1 simultaneously transitions from state NEW to state OLD in time step 4. The optimal action is to replace this component. This is because components 1 and 2 are the most important in the system: at least one of them is necessary to satisfy all demands. Therefore the system proactively replaces the component in state OLD. Since component 2 is still in state NEW and operational all demands can be satisfied even if component 2 is in state REPLACING and therefore non-operational. In the right column, components 2 and 4 transition to state FAILED in time step 4. The optimal action is REPLACE for both components. In this case the immediate replacement is more beneficial then the spreading observed earlier. Since there are already different remaining lifetimes for the components there is no need for additional spreading. Further the other components could transition to state OLD. The system is very likely to replace components in state OLD. Having more operational components increases the probability of replacing components in state OLD without incurring the penalty for not satisfying any demands.

Case 2: all components are OLD

Figure 5.4 shows the optimal policy and consecutive states if initially all six components are in state OLD. We observe that in the first time step the optimal action is to REPLACE components 1, 3 and 6. This seems counterintuitive. Replacing components 1 and 2 in the same time step isolates node 2 from the network. Therefore the system willingly incurs a penalty for unsatisfied demand at this node. However, this action decreases the probability of future failures. In all consecutive time steps the system can satisfy all demands while replacing the remaining components still in state OLD. Thus, in this scenario, it is better to incur a smaller penalty in the short term than to potentially incur a larger penalty in the future.



Optimal repair and replacement actions, along with resulting states, when all six components are initially in state OLD. Left column: the optimal replacement schedule in the absence of any other events. Right column: if component 4 transitions to state FAILED, the replacement of that component is more important than the replacement of components 2 and 5.

Figure 5.4: Optimal Policy if all Components are OLD

5.3 Parametric Analysis of the Fuel System

Next we vary the parameters for the fuel system to analyze different scenarios and effects of changes to input parameters. These scenarios show the possible use of the multiple-component model beyond the identification of optimal repair and replacement policies.

5.3.1 The Value of Information

In the first set of scenarios we analyze the value of information. We consider two cases. Under *complete information*, we assume that the decisions to take either the action REPAIR or REPLACE are based on complete knowledge of each component's state (e.g., whether a component is in state NEW, OLD or FAILED). In contrast, under *incomplete information*, we assume that that the decision to take either the action REPAIR or REPLACE are based solely on the information that a component is operational (e.g., in states NEW or OLD) or non-operational (e.g., in state FAILED). The implementation in the linear program is straight forward. In the case of incomplete information the proactive replacement of a component is infeasible. Therefore the state-action pair (OLD, REPLACE) can not occur. We fix all decision variables containing this state-action pair to zero. So the system is prohibited from spending any proportion of time with a proactive replacement. We compare the performance of the fuel system with complete and incomplete information. We fix the transition probabilities for all components ($\alpha = 0.2, \beta = 0.1, \gamma = 0.3$), and vary c_1, c_2.

Table 5.2: The Value of Information

		complete information		incomplete information	
c_1	c_2	cost	avg. worker	cost	avg. worker
1	2	15.738	2.115	17.529	1.585
1	3	16.784	2.066	18.321	1.583
2	3	16.786	2.065	18.321	1.584
2	4	17.799	2.00	19.113	1.582

This table shows the long-run average cost and long-run average number of workers in use per time step for complete and incomplete information and for different values of c_1, c_2.

Table 5.2 shows the long-run average cost and long-run average number of workers in use per time step for a system with complete and incomplete information and for different values of c_1 and c_2. First, we can see for identical c_1 and c_2 the system with incomplete information is more costly compared to the system with complete information. The system with complete information benefits from the possibility of proactive replacements. Second, the system with complete information uses on average more workers per time step compared to the system with incomplete information. Proactive replacement of components requires additional workers whereas the system with incomplete information has less need for workers since it can only use the actions REPAIR and reactive REPLACE. Therefore fewer workers are required compared to the system with full information, since here proactive REPLACE action for old components is an additional option. Third, with increasing c_1 and c_2, the average cost per time step also increases. Since the transition probabilities are identical the components will fail with the same probability. The necessary REPAIR and REPLACE actions are more costly. Therefore the average cost per time step increases as well. Fourth, the average number of workers per time step depends on the ratio of c_1 and c_2. If the REPLACE action is relatively costly compared to the REPAIR action the

Table 5.3: “Full-Time” Workers

	complete information		incomplete information	
workers	cost	utilization	cost	utilization
1	26.341	0.930	26.341	0.930
2	16.685	0.936	18.612	0.735
4	15.756	0.528	17.578	0.394
6	15.738	0.352	17.530	0.265
10	15.738	0.211	17.529	0.158

This table shows effect of different number of “full-time workers” on the long-run average cost and long-run average utilization for systems with complete and incomplete information.

average number of workers is lower compared to cases with similar c_1 and c_2. If the REPLACE action is relatively costly the system favors REPAIRS action. Since REPAIR actions require less workers then REPLACE actions, the average number of workers per time step is lower.

5.3.2 The Value of Full-Time Workers

In the previous analysis, the assumption is that we could “hire” an infinite amount of workers for a cost of c_1 or c_2 to conduct REPAIR or REPLACE actions respectively. In this scenario we examine a fixed number of full-time workers. Since we don’t “hire” external workers $c_1 = c_2 = 0$. We fix the transitions probabilities ($\alpha = 0.2, \beta = 0.1, \gamma = 0.3$), and vary the number of workers and complete and incomplete information.

Table 5.3 shows the average cost and average utilization of each worker per time step for systems with complete and incomplete Information. Recall, that the system requires one worker for every REPAIR action and two workers for every REPLACE action. Further recall, that we set $c_1 = c_2 = 0$. We assume the cost of parts, etc. are zero, so c_1, c_2 are the only cost to “hire” a worker. Therefore the cost will be lower compared to the scenarios with "hired" workers. Since we consider “full-time workers”, they will be either “busy” conducting a REPAIR or REPLACE action or be “idle” (i.e., conduct the NONE action). To capture this, we define *utilization* to be the long-run average proportion of time an individual workers conducts a REPAIR or

REPLACE action.

First, with exception of one worker the average cost per time step is higher for the system with incomplete information compared to the system with complete information. The utilization is lower for the system with incomplete information. These observations correlate with our findings of the value of information.

Second, the cost and utilization is the same for the scenario with just one worker. In this scenario the system can not apply REPLACE actions, since it has only one available worker. Therefore the system can not benefit from the complete information.

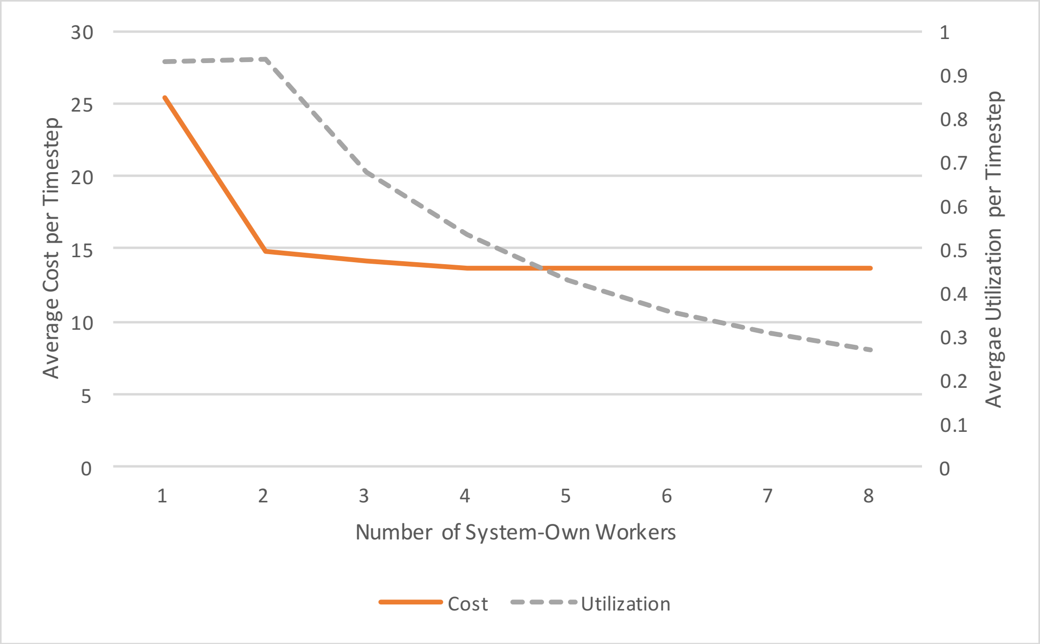
Third, to perform all possible actions the system needs twelve workers. Figures 5.5 and 5.6 depict the long-run average cost for different numbers of workers for complete and incomplete information respectively. In both cases we can see, that the decrease in cost is significant from 1 to 2 workers. For additional workers the decrease in cost is diminishing. The same is true for the utilization but less distinctive.

Fourth, from Figure 5.5 we see that the utilization increases slightly from 1 to 2 workers and then decreases continuously for a system with complete information. With just one worker only REPAIR actions are possible. Therefore the proactive REPLACE action for OLD components is not possible. The system needs a minimum of two workers to get this option of proactive actions. Therefore the utilization spikes for two workers. With additional workers the utilization drops since more workers can share the load. Fifth, Figure 5.6 depicts the utilization for a system with incomplete information. The utilization decreases continuously, since more workers can are available to perform the actions.

5.3.3 The Value of More Reliable Components

In this scenario we vary the transition probabilities α, β and γ. The cost for REPAIR and REPLACE actions are fixed to $c_1 = 1, c_2 = 2$.

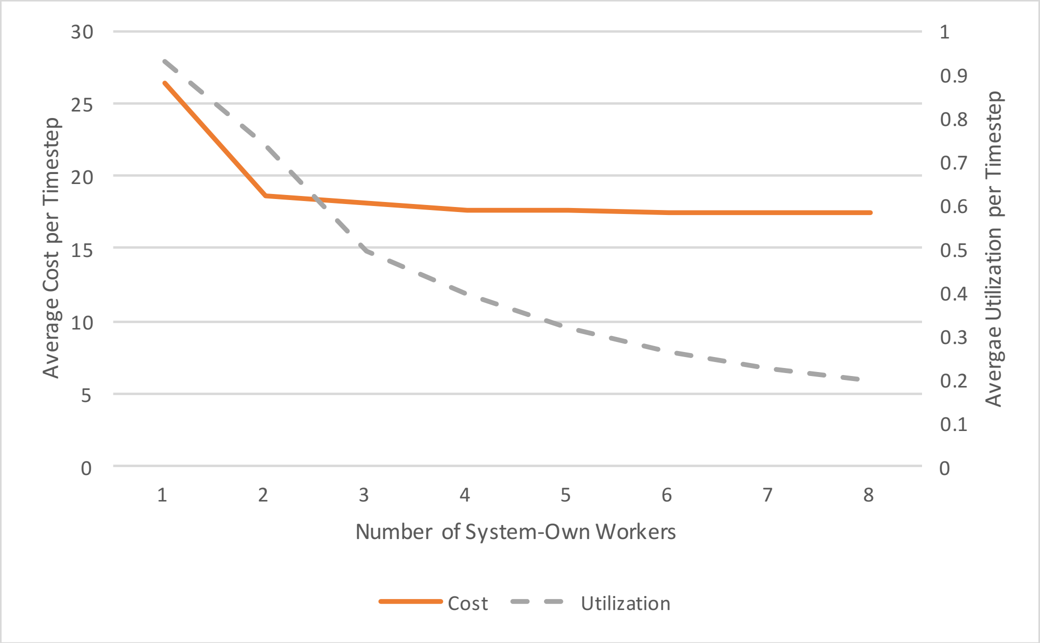
Table 5.4 shows the long-run average cost and number of workers in use per time step for system with complete and incomplete information for varying α, β, γ. First, we can see the general trends for cost and utilization in a system with complete and incomplete information also hold for different transition probabilities. Second, the



For an increasing number of workers the long-run average cost and utilization per time step decreases. This improvement of cost and utilization diminishes for large number of workers

Figure 5.5: Long-Run Average Cost and Utilization for a Fuel System with Full-Time Workers and Complete Information.

long-run average cost and number of average number of workers in use per time step grows with increasing transition probabilities. Higher transition probabilities mean a higher rate of failure. Since the components are more likely to fail, more REPAIR and REPLACE actions are required to keep the system operational. Therefore the system will induce higher costs. Third, for lower transition probabilities the difference in cost for systems with complete and incomplete information diminishes. If components in state OLD are less likely to fail, the benefit of proactive REPLACE actions gets smaller. The system is takes the risk of failure instead of setting the component to non-operational for the proactive REPLACE action. If the number of proactive REPLACE action gets smaller the benefit of complete information gets smaller as well. Therefore complete information has an higher impact for system with higher rates of failure.



For an increasing number of workers the long-run average cost and utilization per time step decreases. This improvement of cost and utilization diminishes for large number of workers

Figure 5.6: Long-Run Average Cost and Utilization for a Fuel System with Full-Time Workers and Incomplete Information.

Table 5.4: Varying Transition Probabilities

			complete information		incomplete information	
α	β	γ	cost	avg. worker	cost	avg. worker
0.02	0.01	0.03	8.536	0.211	8.537	0.205
0.10	0.05	0.15	11.591	1.241	12.076	0.915
0.20	0.10	0.30	15.738	2.115	17.529	1.585
0.30	0.20	0.50	21.633	2.777	24.759	2.226

This table shows the long-run average cost and number of workers in use per time step decreases if the transition probabilities decrease (i.e., “better components”). For more reliable components the difference between system with complete and incomplete information gets smaller.

CHAPTER 6: Conclusions and Future Work

6.1 Conclusions

The Multiple-Component Model is suitable to study of optimal repair and replacement policies for any system with n independent components that are prone to age-based failure. Due to its generic nature we can apply the model to various systems of multiple components. The fuel system is just one of many possible examples. The model solution provides an optimal repair and replacement strategy for every possible state of the system. Such a policy allows an operator to maintain his system optimally.

We can extend or modify the linear program used to solve the Markov decision process to capture different scenarios. The option to add additional constraints and change the reward structure of the objective function additionally highlights the generic nature of this model.

The Multiple-Component Model can provide additional insights as well. As illustrated in this thesis, the results of parametric analyses can provide additional information on the consequences of potential changes to the system, such as the value of using more reliable components. Moreover, it is possible to adjust the linear program explicitly to consider additional constraints, such as the limited availability of repair crews. All of these modifications can aid in policy development.

6.2 Future Work

The formulation and analysis of the Multiple-Component Model depends on several assumptions that help to make the analysis simpler and the computation more tractable. The model is time-discrete, and the representation of component age has been simplified to just "NEW" and "OLD". In addition, components are assumed to age and fail independently of one another. A higher resolution representation of system age and more realistic dependencies would significantly increase the complexity and the computational burden to solve the linear program. It would also increase the input data requirements for running a model. In its current form, we can solve a system with seven components in a reasonable time on a personal computer. With additional computational power it should be possible to solve the model for system of more the seven components and/or to increase the resolution of the model.

Future work should consider ways of increasing the realism (and applicability) of the model, while also looking to decrease the computational burden to solve it. For example, we know that increasing model realism often requires additional constraints for the linear program. These constraints could render some policies to be infeasible and thereby actually reduce the number of feasible solutions.

Another avenue for future work could be to identify ways in which a larger system can be decomposed into smaller sub-systems that can be solved independently. Such ideas could take advantage of the growing power of parallel computing platforms, and thereby greatly increase the scale of systems that can be studied in practice.

Finally, in situations where exact solutions to large systems are not obtainable, it could be worthwhile to look for suitable heuristic approaches that obtain solutions that are "good enough" given limited computing resources.

APPENDIX A:
Results from the Linear Program

A.1 Two-Component Model

Two components with identical parameters:

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4, r = 5, \qquad c_1 = 1 \qquad c_2 = 2$$

```
Objective value is 5.826087
Variable X
{('1', '1'): 0.08401596303297629,
 ('1', '2'): 0.10501995379122034,
 ('1', '3'): 0.0,
 ('1', '4'): 0.0,
 ('1', '5'): 0.05040957781978575,
 ('1', '6'): 0.0,
 ('1', '7'): 0.0,
 ('1', '8'): 0.05040957781978576,
 ('2', '1'): 0.10501995379122034,
 ('2', '2'): 0.13127494223902536,
 ('2', '3'): 0.0,
 ('2', '4'): 0.0,
 ('2', '5'): 0.06301197227473218,
 ('2', '6'): 0.0,
 ('2', '7'): 0.0,
 ('2', '8'): 0.06301197227473218,
 ('3', '1'): 0.0,
 ('3', '2'): 0.0,
 ('3', '3'): 0.0,
 ('3', '4'): 0.0,
 ('3', '5'): 0.0,
 ('3', '6'): 0.0,
 ('3', '7'): 0.0,
 ('3', '8'): 0.0,
 ('4', '1'): 0.0,
 ('4', '2'): 0.0,
 ('4', '3'): 0.0,
 ('4', '4'): 0.0,
 ('4', '5'): 0.0,
 ('4', '6'): 0.0,
 ('4', '7'): 0.0,
 ('4', '8'): 0.0,
 ('5', '1'): 0.05040957781978576,
 ('5', '2'): 0.06301197227473218,
 ('5', '3'): 0.0,
 ('5', '4'): 0.0,
 ('5', '5'): 0.03024574669187147,
 ('5', '6'): 0.0,
 ('5', '7'): 0.0,
 ('5', '8'): 0.030245746691871446,
 ('6', '1'): 0.0,
 ('6', '2'): 0.0,
 ('6', '3'): 0.0,
 ('6', '4'): 0.0,
 ('6', '5'): 0.0,
 ('6', '6'): 0.0,
 ('6', '7'): 0.0,
 ('6', '8'): 0.0,
 ('7', '1'): 0.0,
 ('7', '2'): 0.0,
 ('7', '3'): 0.0,
 ('7', '4'): 0.0,
 ('7', '5'): 0.0,
 ('7', '6'): 0.0,
  ('7', '7'): 0.0,
  ('7', '8'): 0.0,
  ('8', '1'): 0.050409577819785764,
  ('8', '2'): 0.06301197227473217,
  ('8', '3'): 0.0,
  ('8', '4'): 0.0,
  ('8', '5'): 0.030245746691871453,
  ('8', '6'): 0.0,
  ('8', '7'): 0.0,
  ('8', '8'): 0.03024574669187147}
```

A.2 Two-Component-Model with 1 Worker Constraint

Two components with identical parameters:

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4, r = 5, \qquad c_1 = 1 \qquad c_2 = 2$$

```
Objective value is 4.999271
Variable X
    ('5', '8') 0
    ('2', '2') 0.27352297593
    ('8', '3') 0.0
    ('6', '7') 0.0
    ('5', '5') 0
    ('8', '8') 0.0
    ('1', '6') 0.0
    ('7', '2') 0.132101466894
    ('3', '7') 0.0
    ('2', '7') 0.11589269795
    ('8', '4') 0.0
    ('7', '6') 0.0
    ('3', '4') 0.0
```

```
('1', '2') 0.0
('4', '4') 0
('7', '5') 0
('2', '3') 0.0
('4', '6') 0.0
('4', '7') 0.0966042629062
('3', '8') 0.0
('1', '7') 0.0
('5', '4') 0
('4', '3') 0.0
('7', '1') 0.0
('2', '8') 0.0
('8', '5') 0
('3', '3') 0.0
('5', '3') 0
('7', '4') 0.0463570791798
('2', '4') 0.166139881676
('8', '1') 0.0
('1', '1') 0.0
('6', '1') 0.0
('5', '7') 0
('6', '2') 0.0
('7', '8') 0.0
('1', '5') 0
('4', '5') 0
('6', '4') 0.0
('6', '5') 0
('3', '2') 0.0
('5', '2') 0
('2', '6') 0.0
('2', '5') 0
('6', '8') 0.0
('4', '1') 0.0
```

```
    ('8', '6') 0.0
    ('3', '6') 0.0
    ('4', '8') 0.0
    ('5', '6') 0
    ('7', '7') 0.0
    ('2', '1') 0.0
    ('8', '2') 0.0
    ('6', '6') 0.0
    ('1', '3') 0.0
    ('3', '1') 0.0
    ('7', '3') 0.0437636761488
    ('1', '8') 0.0
    ('1', '4') 0.0
    ('4', '2') 0.125617959316
    ('8', '7') 0.0
    ('3', '5') 0
    ('6', '3') 0.0
    ('5', '1') 0
```

A.3 Two-Component-Model with 2 Worker Constraint

Two components with identical parameters:

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4, r = 5, \qquad c_1 = 1 \qquad c_2 = 2$$

```
Objective value is 5.774964
Variable X
    ('5', '8') 0.0213501911569
    ('2', '2') 0.0
    ('8', '3') 0.0
    ('6', '7') 0.0
    ('5', '5') 0
```

```
('8', '8') 0.0
('1', '6') 0.0
('7', '2') 0.0
('3', '7') 0.0
('2', '7') 0.0
('8', '4') 0.0
('7', '6') 0.0
('3', '4') 0.0
('1', '2') 0.123780594901
('4', '4') 0.0
('7', '5') 0.0
('2', '3') 0.0
('4', '6') 0.0
('4', '7') 0.0
('3', '8') 0.01070287741
('1', '7') 0.0
('5', '4') 0
('4', '3') 0.0
('7', '1') 0.0
('2', '8') 0.0356784321344
('8', '5') 0.0671050732157
('3', '3') 0.0
('5', '3') 0.0
('7', '4') 0.0
('2', '4') 0.0
('8', '1') 0.0687614965145
('1', '1') 0.0655666495237
('6', '1') 0.0
('5', '7') 0.0
('6', '2') 0.0
('7', '8') 0.0
('1', '5') 0.0430802921783
('4', '5') 0
```

```
('6', '4') 0.0
('6', '5') 0.0
('3', '2') 0.0
('5', '2') 0.0560918105155
('2', '6') 0.0917457442415
('2', '5') 0.103617132102
('6', '8') 0.0
('4', '1') 0.0
('8', '6') 0.0
('3', '6') 0.0
('4', '8') 0.0
('5', '6') 0.0
('7', '7') 0.0
('2', '1') 0.127399677967
('8', '2') 0.0361851037637
('6', '6') 0.0
('1', '3') 0.0
('3', '1') 0.0
('7', '3') 0.0
('1', '8') 0.0543252525535
('1', '4') 0.0
('4', '2') 0.0
('8', '7') 0.0
('3', '5') 0.0
('6', '3') 0.0
('5', '1') 0.0946096718215
```

A.4 Two-Component-Model with 3 Worker Constraint

Two components with identical parameters:

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4, r = 5, \qquad c_1 = 1 \qquad c_2 = 2$$

```
Objective value is 5.788291
Variable X
    ('5', '8') 0.0223668600677
    ('2', '2') 0.0
    ('8', '3') 0.0
    ('6', '7') 0.0
    ('5', '5') 0
    ('8', '8') 0.0
    ('1', '6') 0.0
    ('7', '2') 0.0
    ('3', '7') 0.0
    ('2', '7') 0.0
    ('8', '4') 0.0
    ('7', '6') 0.0
    ('3', '4') 0.0
    ('1', '2') 0.117807949618
    ('4', '4') 0.0
    ('7', '5') 0.0
    ('2', '3') 0.0
    ('4', '6') 0.0
    ('4', '7') 0.0
    ('3', '8') 0.0
    ('1', '7') 0.0
    ('5', '4') 0.0
    ('4', '3') 0.0
    ('7', '1') 0.0
    ('2', '8') 0.0350580531748
    ('8', '5') 0.0564601980987
    ('3', '3') 0.0
    ('5', '3') 0.0
    ('7', '4') 0.0
    ('2', '4') 0.0
    ('8', '1') 0.0706204307414
```

```
('1', '1') 0.061096959977
('6', '1') 0.0
('5', '7') 0.0
('6', '2') 0.0
('7', '8') 0.0109510310861
('1', '5') 0.0435952455564
('4', '5') 0.0109510310861
('6', '4') 0.0
('6', '5') 0.0
('3', '2') 0.0
('5', '2') 0.058267440861
('2', '6') 0.0928247011415
('2', '5') 0.103265134451
('6', '8') 0.0
('4', '1') 0.0
('8', '6') 0.0
('3', '6') 0.0
('4', '8') 0.0
('5', '6') 0.0
('7', '7') 0.0
('2', '1') 0.14069358754
('8', '2') 0.0382620424839
('6', '6') 0.0
('1', '3') 0.0
('3', '1') 0.0
('7', '3') 0.0
('1', '8') 0.0530709637219
('1', '4') 0.0
('4', '2') 0.0
('8', '7') 0.0
('3', '5') 0.0
('6', '3') 0.0
('5', '1') 0.0847083703953
```

A.5 Two-Component-Model with 4 Worker Constraint

Two components with identical parameters:

$$\alpha = 0.5, \qquad \beta = 0.1, \qquad \gamma = 0.4, r = 5, \qquad c_1 = 1 \qquad c_2 = 2$$

```
Objective value is 5.826087
Variable X
    ('5', '8') 0.0302457466919
    ('2', '2') 0.131274942239
    ('8', '3') 0.0
    ('6', '7') 0.0
    ('5', '5') 0.0302457466919
    ('8', '8') 0.0302457466919
    ('1', '6') 0.0
    ('7', '2') 0.0
    ('3', '7') 0.0
    ('2', '7') 0.0
    ('8', '4') 0.0
    ('7', '6') 0.0
    ('3', '4') 0.0
    ('1', '2') 0.105019953791
    ('4', '4') 0.0
    ('7', '5') 0.0
    ('2', '3') 0.0
    ('4', '6') 0.0
    ('4', '7') 0.0
    ('3', '8') 0.0
    ('1', '7') 0.0
    ('5', '4') 0.0
    ('4', '3') 0.0
    ('7', '1') 0.0
    ('2', '8') 0.0630119722747
```

```
('8', '5') 0.0302457466919
('3', '3') 0.0
('5', '3') 0.0
('7', '4') 0.0
('2', '4') 0.0
('8', '1') 0.0504095778198
('1', '1') 0.084015963033
('6', '1') 0.0
('5', '7') 0.0
('6', '2') 0.0
('7', '8') 0.0
('1', '5') 0.0504095778198
('4', '5') 0.0
('6', '4') 0.0
('6', '5') 0.0
('3', '2') 0.0
('5', '2') 0.0630119722747
('2', '6') 0.0
('2', '5') 0.0630119722747
('6', '8') 0.0
('4', '1') 0.0
('8', '6') 0.0
('3', '6') 0.0
('4', '8') 0.0
('5', '6') 0.0
('7', '7') 0.0
('2', '1') 0.105019953791
('8', '2') 0.0630119722747
('6', '6') 0.0
('1', '3') 0.0
('3', '1') 0.0
('7', '3') 0.0
('1', '8') 0.0504095778198
```

```
('1', '4') 0.0
('4', '2') 0.0
('8', '7') 0.0
('3', '5') 0.0
('6', '3') 0.0
('5', '1') 0.0504095778198
```

APPENDIX B:
Results from Application to a Fuel System

B.1 Reward Structure for a Five-Component Fuel-system (Excerpt)

```
('1', '4', '6', '5', '4'): 27.0,
('1', '4', '6', '5', '5'): 28.0,
('1', '4', '6', '5', '6'): 26.0,
('1', '4', '6', '5', '7'): 26.0,
('1', '4', '6', '5', '8'): 26.0,
('1', '4', '6', '6', '1'): 24.0,
('1', '4', '6', '6', '2'): 24.0,
('1', '4', '6', '6', '3'): 24.0,
('1', '4', '6', '6', '4'): 25.0,
('1', '4', '6', '6', '5'): 26.0,
('1', '4', '6', '6', '6'): 24.0,
('1', '4', '6', '6', '7'): 24.0,
('1', '4', '6', '6', '8'): 24.0,
('1', '4', '6', '7', '1'): 24.0,
('1', '4', '6', '7', '2'): 24.0,
('1', '4', '6', '7', '3'): 24.0,
('1', '4', '6', '7', '4'): 25.0,
('1', '4', '6', '7', '5'): 26.0,
('1', '4', '6', '7', '6'): 24.0,
('1', '4', '6', '7', '7'): 24.0,
('1', '4', '6', '7', '8'): 24.0,
('1', '4', '6', '8', '1'): 24.0,
('1', '4', '6', '8', '2'): 24.0,
('1', '4', '6', '8', '3'): 24.0,
('1', '4', '6', '8', '4'): 25.0,
('1', '4', '6', '8', '5'): 26.0,
('1', '4', '6', '8', '6'): 24.0,
('1', '4', '6', '8', '7'): 24.0,
('1', '4', '6', '8', '8'): 24.0,
('1', '4', '7', '1', '1'): 24.0,
('1', '4', '7', '1', '2'): 24.0,
('1', '4', '7', '1', '3'): 24.0,
('1', '4', '7', '1', '4'): 25.0,
('1', '4', '7', '1', '5'): 26.0,
('1', '4', '7', '1', '6'): 24.0,
('1', '4', '7', '1', '7'): 24.0,
('1', '4', '7', '1', '8'): 24.0,
('1', '4', '7', '2', '1'): 24.0,
('1', '4', '7', '2', '2'): 24.0,
('1', '4', '7', '2', '3'): 24.0,
('1', '4', '7', '2', '4'): 25.0,
('1', '4', '7', '2', '5'): 26.0,
('1', '4', '7', '2', '6'): 24.0,
('1', '4', '7', '2', '7'): 24.0,
('1', '4', '7', '2', '8'): 24.0,
('1', '4', '7', '3', '1'): 32.0,
('1', '4', '7', '3', '2'): 32.0,
```

List of References

Agrawal, A., R.E. Barlow. 1984. A survey of network reliability and domination theory. *Operations Research* **32**(3) 478–492. URL http://dx.doi.org/10.1287/opre.32.3.478.

Alderson, D., G. Brown, W.M. Carlyle. 2015. Operational models of infrastructure resilience. *Risk Analysis* **35**(4) 562–586.

Barlow, R.E., L.C. Hunter. 1961. Reliability analysis of a one-unit system. *Operations Research* **9**(2) 200–208. URL http://dx.doi.org/10.1287/opre.9.2.200.

Derman, C. 1963a. On optimal replacement rules when changes of state are Markovian. R. Bellman, E. Richard, eds., *Mathematical optimization techniques*. University of California Press, Berkeley, 201–210.

Derman, C. 1963b. Optimal replacement and maintenance under markovian deterioration with probability bounds on failure. *Management Science* **9**(3) 478–481.

Dogramaci, A., N.M. Fraiman. 2004. Replacement decisions with maintenance under uncertainty: An embedded optimal control model. *Operations Research* **52**(5) 785–794. URL http://dx.doi.org/10.1287/opre.1040.0133.

Hart, W.E., C. Laird, J. Watson, D.L. Woodruff. 2012. *Pyomo–optimization modeling in python*, vol. 67. Springer Science & Business Media.

Hart, W.E., J. Watson, D.L. Woodruff. 2011. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation* **3**(3) 219–260.

Hillier, F.S., G.J. Lieberman. 2010. *Introduction to operations research*. 9th ed. McGraw-Hill Higher Education, New York.

IBM. 2016. IBM ILOG CPLEX Optimization Studio Community Edition. URL http://www-01.ibm.com/software/websphere/products/optimization/cplex-studio-community-edition/. Accessed March 13, 2016.

Lariviere, M.A. 2006. A note on probability distributions with increasing generalized rates. *Operations Research* **54**(3) 602–604. URL http://dx.doi.org/10.1287/opre.1060.0282.

Love, C.E., Z.G. Zhang, M.A. Zitron, R. Guo. 2000. A discrete semi-markov decesion model to determine the optimal repair/replacement policy under general repairs. *European Journal of Operations Research* **125** 398–409. URL www.elsevier.com/locate/dsw.

McCall, J.J. 1965. Maintenance policies for stochastically failing equipment: A survey. *Management Science* **11**(5) 493–524. URL http://dx.doi.org/10.1287/mnsc.11.5.493.

Pierskalla, W.P., J.A. Voelker. 1976. A survey of maintenance models: The control and surveillance of deteriorating systems. *Naval Research Logistics Quarterly* **23**(3) 358–388.

PSF. 2016. Python language reference. URL http://www.python.org. Version 2.7.

Puterman, M.L. 2005. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 1st ed. Wiley-Interscience.

Rajagopalan, S. 1998. Capacity expansion and equipment replacement: A unified approach. *Operations Research* **46**(6) 846–657. URL http://dx.doi.org/10.1287/opre.46.6.846.

Ross, S.M. 2014. *Introduction to Probability Models*. 11th ed. Academic Press.

Sherif, Y.S., M.L. Smith. 1981. Optimal maintenance models for systems subject to failure— a review. *Naval Research Logistics Quarterly* **28**(1) 47–74.

Sim, S.H., J. Endrenyi. 1993. A failure-repair model with minimal and major maintenance. *IEEE Transactions on Reliability* **42**(1) 134–140.

The White House. 2015. Presidential proclamation – critical infrastructure security and resilience month, 2015. URL https://www.whitehouse.gov/the-press-office/2015/10/29/presidential-proclamation-critical-infrastructure-security-and.

Thomas, L.C., D.P. Gaver, P.A. Jacobs. 1987. Optimal inspection policies for standby systems. *Stochastic Models* **3** 259–274.

von Noortwijk, J.M., D.M. Frangopol. 2014. Two probabilistic life-cycle maintenance models for deteriorating civil infrastructure. *Probabilistic Enginering Mechanics* **19** 345–359.

Zhang, Y.L. 2002. A geometric-process repair-model with good-as-new repair. *IEEE Transactions on Reliability* **51**(2) 223–228.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California